DOCUMENT RESUME

ED 078 676 EM 011 223

AUTHOR Grossel, Roger Louis

TITLE A Computer-Based Education Approach to Electrical Network Theory: Lesson Development, Use and Evaluation.

INSTITUTION Illinois Univ., Urbana. Computer-Based Education Lab.

REPORT NO CERL-R-X-29

PUB DATE Jul 71

NOTE 81p.; Ph.D. Thesis, University of Illinois

EDRS PRICE MF-$0.65 HC-$3.29

DESCRIPTORS Academic Achievement; College Students; *Computer Assisted Instruction; Computer Programs; Course Evaluation; Doctoral Theses; *Engineering Education, Higher Education; *Instructional Design; Programing; *Teaching Methods

IDENTIFIERS Electrical Network Theory; Kirchoffs Laws; *PLATO III Computer Assisted Instruction System; TUTOR Language

ABSTRACT

This study investigated the use of the PLATO III computer assisted instruction (CAI) system to assist in the teaching of an electrical network theory course. It sought to: 1) identify the topics and teaching strategies most amenable to CAI system; 2) develop computer programs needed to teach those topics; 3) use, evaluate and service the programs; and 4) evaluate the overall effectiveness of this research. Using the TUTOR language, the researcher developed a network drawing utility routine, wrote a lesson on Kirchhoff's laws, and developed a DC analysis version of the on-line network analyzer. Two one-semester sections of the course were taught with students having 20% of their instructional time on the CAI system. Results which showed that student attendance and achievement were high and their attitudes positive led to the conclusions that network theory can be taught by CAI systems and that TUTOR is appropriate language. Further implications were that future research should study additional instruction instructional modes and investigate the utility of interstation communication, pre-recorded feedback, and the information structure oriented approach. (PB)

CERL REPORT X-29 JULY, 1971

A COMPUTER-BASED EDUCATION APPROACH TO ELECTRICAL NETWORK THEORY: LESSON DEVELOPMENT, USE AND EVALUATION

ROGER LOUIS GROSSEL

Computer-based Education Research Laboratory

University of Illinois Urbana Illinois

A COMPUTER-BASED EDUCATION APPROACH TO ELECTRICAL NETWORK THEORY: LESSON DEVELOPMENT, USE AND EVALUATION

BY

ROGER LOUIS GROSSEL
B.S., University of Wisconsin, 1958
M.S., University of Wisconsin, 1964

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1971

Urbana, Illinois

ACKNOWLEDGEMENT

I wish to thank the U.S. Air Force for allowing me to do this research by assigning me to the University of Illinois.

I also wish to acknowledge the continued help and cooperation of the complete staff of the Computer-based Education Research Laboratory of the University of Illinois. It is a pleasure to work with such fine people. Special individual acknowledgement goes to my thesis advisor and Director of the Laboratory, Dr. Donald L. Bitzer.

Other individual thanks are due to Dr. John P. Gordon and Mr. David V. Meller for their many interesting discussions and their helpful advice, to Jack Gladin who did the photographic work for this thesis, and to the 19 students who took the PLATO sections of EE260.

Finally, I wish to thank the two women in my life, Joan and Jo-Ann, for their part.

PREFACE

Faced with demands for both quantity and quality, educators have attempted to introduce modern technology. For over a decade now, numerous efforts have been made to apply computer technology to new instructional modes which use time-sharing and student interaction with the subject matter. These instructional modes tend to be highly individualized to the student and not purely computational. The term "computer-assisted instruction" (CAI) has been given to these diverse efforts.

The PLATO program at the University of Illinois is one such effort that has taken a broad systems approach to CAI. This approach consisted of new hardware (deemed necessary for economic viability), software, and educational research-all developed together as a system. It has come to be categorized as "computer-based education".

This research investigates the potential role of a computer-based system in the teaching of introductory network theory. It attempts to partially answer the question, "What is educationally possible or desirable?" It does not propose CAI as the most effective or efficient medium, nor does it compare CAI to other media. It does, however, attempt to find out some of the CAI characteristics and capabilities, for use when media selections are made.

The approach used was to start with the PLATO III system and an existing network theory course. Consideration of the course topics, the various teaching strategies, and the PLATO III system capabilities led to the selection of several topics and teaching strategies, and eventually to the development of lessons (computer programs) to assist in teaching those topics. Some rationale for developing the lessons is presented in Section 4. The actual lessons that were developed are discussed in Section 5, along with some results from their use.

Development, use, evaluation, and revision of the lessons took place over the two semesters of the 1970-71 school year. A total of 19 students took the special course for credit. In the second semester, about 30% of the scheduled hours were in the PLATO classroom.

Several benefits from the use of PLATO were observed. First, a high degree of student achievement was obtained by use of lessons combining tutorial and drill and practice instructional modes. Second, the student attitude was found to be highly positive for properly working lessons. It's not often a student says "it's fun" during a school learning experience. Third, the student worked alone at a console, at his own pace, and was actively involved. He learned by doing. One student, recognizing this said "There's no way to sit back and do nothing as in a lecture." Fourth, the teacher was not replaced, he was <u>re</u>placed. During the PLATO classes, he was free to roam about and act as an individual tutor, supplementing the PLATO program differently for each student.

The conclusions and recommendations for further research are included in Section 7.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENT	iii
PREFACE	iv
1. INTRODUCTION	1
1.1 Nature of the Problem	1
1.2 Individualized Instruction	2
1.3 Computer-Based Education	2
2. STATEMENT OF THE PURPOSE	5
3. TWO FUNDAMENTAL INGREDIENTS	6
3.1 The Network Theory Course (EE260)	6
3.2 The Computer-Based Education System (PLATO III)	6
3.2.1 The Hardware	7
3.2.2 The Software	10
4. LESSON PHILOSOPHY AND NEEDS	11
4.1 Utility Programs Needed	11
4.1.1 Drawing of Networks	11
4.1.2 Other Utility Routines	11
4.1.3 Response Evaluation	11
4.2 Teaching Programs	12
4.2.1 Kirchhoff's Laws	12
4.2.2 Simple Networks	13
4.2.3 On-line Network Analyzer	13
4.2.4 Analysis and Simulation of Dynamic Networks	15
4.2.5 Fourier Sums	15
5. EXISTING LESSONS AND SELECTED RESULTS	16
5.1 Utility Routines	16

Page

5.1.1 Drawing of Networks (DRNET) 16

5.1.2 Ordinary Calculations (CALC) 19

5.1.3 Complex Number Operations 24

5.1.4 Student Comments 24

5.1.5 Response Evaluation 26

5.2 Teaching Programs 29

5.2.1 Kirchhoff's Current and Voltage Laws (KCL and KVL) 29

5.2.2 Applications of KCL and KVL 34

5.2.3 Simple Equivalent Networks 36

5.2.4 Nodal Analysis 40

5.2.5 On-line Network Analyzer 40

5.2.6 Analysis and Simulation of Dynamic Networks 51

5.2.7 Natural Response of a Parallel RLC Network. 54

5.2.8 Fourier Sums 57

6. ADDITIONAL RESULTS 59

6.1 The First PLATO Section 59

6.2 The Second PLATO Section 60

6.2.1 Attendance at PLATO Classes 61

6.2.2 Student Attitudes 61

6.2.3 Tutorial Lesson Completion Times for the Class 62

6.2.4 Tutorial Lesson Completion Times for an Individual Student 63

6.2.5 Lesson Development Times 63

Page

7. EVALUATION

7.1 Conclusions 66

7.2 Recommendations 69

LIST OF REFERENCES 71

1. INTRODUCTION

1.1 Nature of the Problem

Something is wrong with education when the top 10% of our high school classes have difficulty in mastering college subjects. Individual differences in learners, coupled with the hierarchal nature of most subject matter, are certainly important factors in the learning of new materials. One central problem with education today appears to be its lack of proper regard for these two factors.

Individual differences in learners is an easily demonstrated fact. Many teachers seem to imply that differences do not exist in a beginning class of students by starting out all students at the same level. It appears incongruous for these same teachers to later attest to these differences by assigning terminal grades according to the normal curve.

The hierarchal nature of most subject matter, too, is easily demonstrated. Much recent research in education has dealt with establishing learning hierarchies for subject matter. Gagne (13) states that the most dependable condition to insure learning is the prior learning of prerequisite capabilities. Koen (17) maintains that we need 100% learning at each step.

If all students are normally distributed with respect to aptitude for a given subject (often met in practice), and if all students receive exactly the same instruction (also often met in practice), then intuitively we might expect the students to be normally distributed in end achievement (which they often are in practice). Carroll (9) alleges that if a normally-distributed-aptitude set of students is provided with instruction appropriate to the characteristics and needs of each student, then the majority of students may be expected to achieve mastery of the subject. His view is that, given enough time, all students can conceivably attain mastery of a given learning task. The problem remains to provide the right kind of instruction so that efficiency of learning is improved and that mastery can be attained.

1.2 Individualized Instruction

Bloom (7) believes that if every student had a very good tutor, most of them would be able to attain mastery of a given subject. Essentially this means that the student must be treated as an individual case, and not one of a random, normally distributed, set of learners.

Individualized instruction is currently being offered as a solution. It has several distinguishing characteristics: the learner proceeds at a self-determined pace; he works at times convenient to him; he begins instruction at a point appropriate to his past achievement; he is provided remedial instruction where necessary; instruction is tailored to fit his special requirements and capabilities; and he has a wide variety of media to choose from (20).

1.3 Computer-Based Education

Educators know, in essence, how to proceed to individualize instruction. The underlying problem to individualizing instruction is financial: we can only individualize to the extent that funds are available. Computer-based education has recently appeared on the scene as an economically viable implementation of individualized instruction (1).

The digital computer memory and branching capabilities are well known. Such a machine, embedded in an educationally oriented system, can be programmed to provide all of the following individualized instructional modes: tutorial, drill and practice, computation, simulation, information retrieval, logical problem solving, inquiry, and testing.

Individualized computer teaching programs can provide continuous interaction between student and subject matter, be sensitive to student learning characteristics, incorporate adaptive features to provide both help and enrichment, and keep account of the student proficiency level. In addition, the system can control external devices, such as movie projectors, slide projectors, audio devices, and even equipment for such things as laboratory experiments.

Most instructional modes are envisioned as one student working alone at a console, * which consists of at least a graphic display device plus a keyset. Thus, in interacting with the subject matter through the program written by the

---

*Easley (10, p.25) has reported that, in some cases, there is a possible advantage to working in pairs.

instructor, he can proceed at his own pace. The program attempts to tailor the instruction to the needs of the individual learner. An important feature is that the student is not passive. Rather, by using the keyset he actively participates and even controls his progress.

2. STATEMENT OF THE PURPOSE

This research investigates the use of a particular computer-based education system (PLATO III) to assist the teaching of a particular electrical network theory course (EE260).

The specific purposes are to:

1) Identify the topics and teaching strategies most amenable to computer-based educational assistance,

2) Identify and develop the computer programs needed to assist in teaching the above topics,

3) Use, evaluate, and revise the above programs, and

4) Evaluate the overall effectiveness of this research as an aid to education.

3. TWO FUNDAMENTAL INGREDIENTS

Some results of this research were necessarily determined in part by two fundamental ingredients that were used: an existing course, EE260, and an existing computer-based education system, PLATO III.

3.1 The Network Theory Course (EE260)

The course that was used is Introductory Network Theory (EE260) of the Department of Electrical Engineering of the University of Illinois at Urbana-Champaign. In this course, networks are restricted to be finite, linear, lumped and time-invariant. Major topics included are Kirchhoff's Laws, Equivalent Circuits, Consequences of Linearity, Controlled Sources, Nodal and Mesh Analysis, Classical Differential Equations and Complete Response, and Sinusoidal Steady State Analysis. It is normally taken by sophomores and juniors, is a 3 credit-hour course, and sections meet for 3 hours per week for a semester.

3.2 The Computer-Based Education System (PLATO III)

This research used the PLATO III system at the Computer-based Education Research Laboratory of the University of Illinois at Urbana-Champaign. (1,6) One use of the system is to present information (stimuli) to the student and to react to his responses. The most visible elements of any system are the student console, the computer, and supporting electronics; these are referred to as the hardware. Any computer's actions are specified for every situation by a set of specifications called a program. In an educational system, programs that specifically teach are generally called lessons. The set of lessons and other programs are referred to as the software.

3.2.1 The Hardware

The hardware consists of 75 student consoles (of which 20 can be used by students or lesson authors at any one time), a CDC 1604 computer, and electronic equipment to support the consoles and to act as an interface between the computer and consoles. A diagram of the hardware is shown in Figure 1. Auxiliary equipment, such as projectors, audio devices and even laboratory and research equipment can be computer-controlled from a console.

A console consists of a television monitor display and a keyset, as shown in Figure 2. The keyset, illustrated in Figure 3, is used by students to make requests and responses and by lesson authors to compose and edit lessons. A console is converted to the "author mode" by typing a codeword. The electronics can super-impose computer-generated graphic information and computer-selected slides on the display.

The CDC 1604 computer was modified to time-share the 20 consoles. Peripheral equipment consists of magnetic tapes to store student data and responses, magnetic disks to store lessons and student records, and a line printer. The author, through the lessons he writes, specifies which student data and responses to store during a PLATO class; that evening the magnetic tapes are sorted by student name and printed for him. Lessons are stored on the magnetic disks to allow lesson authors

Plato III

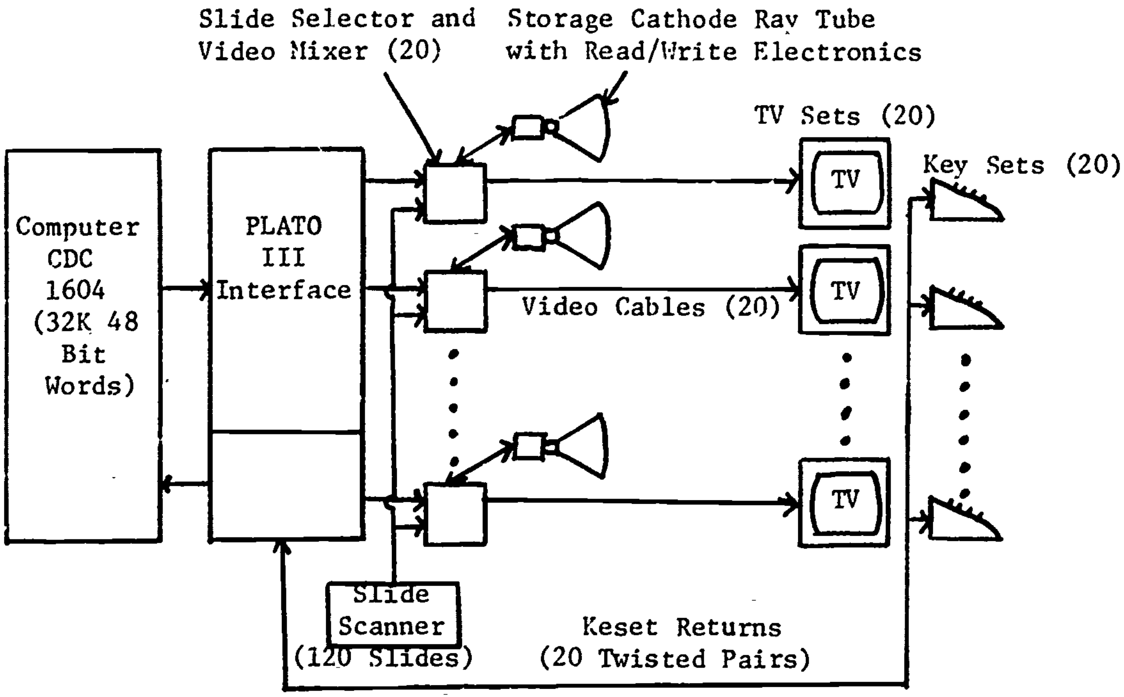


Figure 1. PLATO III Hardware.

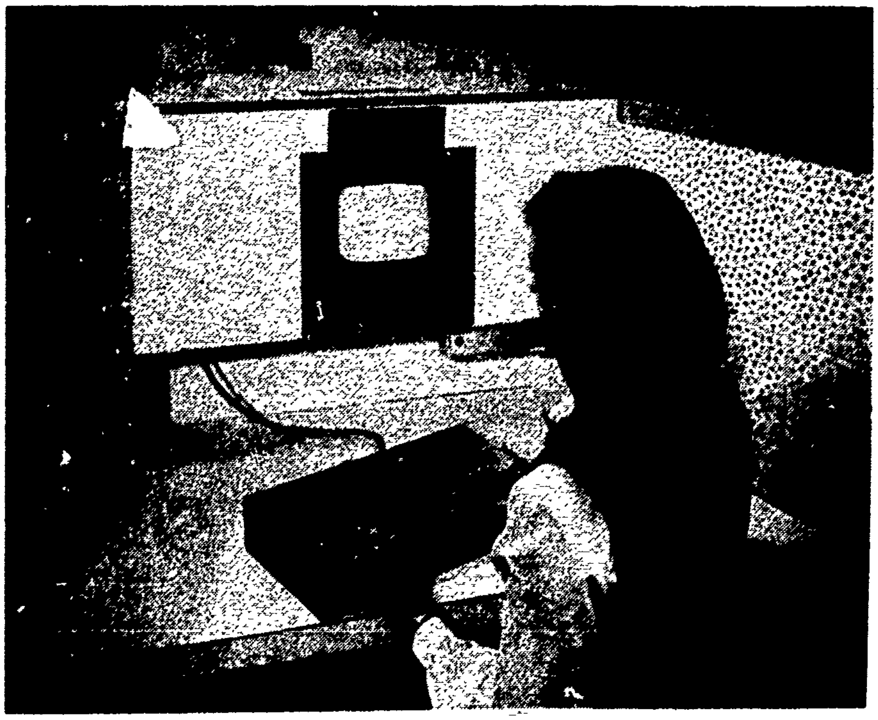


Figure 2. PLATO III Console.

Standard Keyboard

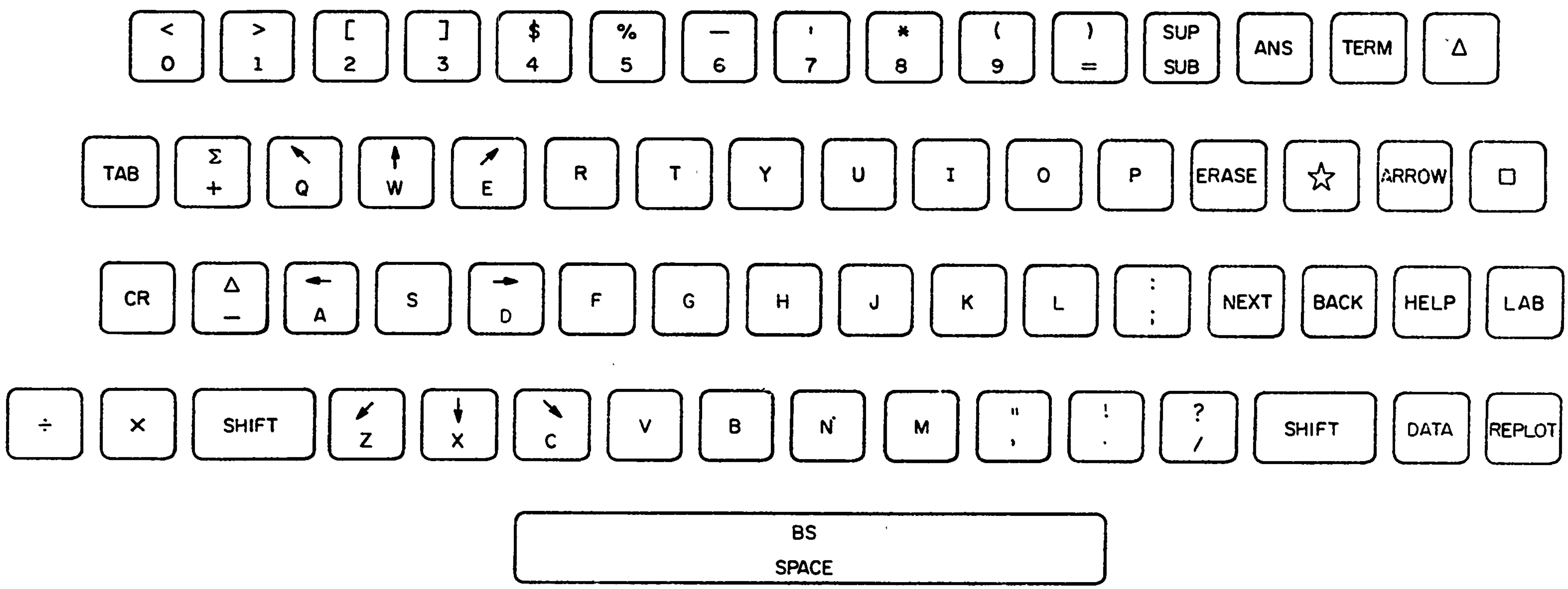


Figure 3.

rapid random access to large amounts of material. Student operation is enabled by directing the system to access the lessons on the disks, compile them into machine code, and store them in the high-speed computer memory. The author institutes a PLATO course by initializing student records on a disk. As each student "signs-in", his records are read into the memory from the disk; as he "signs-off", his current records are stored on the disk. Each subsequent time he "signs-in", he begins the lesson where he left off the last time. The line printer is used for copies of lessons and of student records.

3.2.2 The Software

Lessons for the PLATO III system are written in a computer language called TUTOR, a language designed specifically for a computer-based education system with graphical display (4). TUTOR was designed for use by lesson authors lacking prior experience with computers. It consists of over 80 commands, each of which designates a system-level routine to do complicated things that are useful in an educational system. For example, to have the sentence "Type the v-i relation." written on the student display, the lesson author types the command WRITE, and then the sentence. A system-level routine then does all the work required to display that sentence to a student. Useful lessons can be written using less than 20 commands. Figure 3 indicates only part of the characters available. The complete alphanumeric character set contains over 250 characters.

4. LESSON PHILOSOPHY AND NEEDS

4.1 Utility Programs Needed

4.1.1 Drawing of Networks

An obvious utility program needed was one that could, upon specification of a small set of commands, draw a network of moderate complexity anywhere on the student display. The program had to have the capability to display (or not to display): node and component voltage symbols, current symbols, reference orientations, node numbers, branch numbers, element symbols (primarily "black boxes", resistors, capacitors, and inductors), source symbols, and wires. A tradeoff existed between network density and display size. Too dense a network (elements drawn close together) was not readable; too sparse a network didn't allow enough elements to fit on the display. No graphical symbols for electrical elements were available. Thus these symbols had to be designed.

4.1.2 Other utility Routines

Another need was to have a calculator routine available to the student at all times. Also, a special utility program for elementary operations with complex numbers was needed. In addition, the student was to be able to comment on the lesson at any time. The comments were to be recorded and used later for lesson improvement and for delayed feedback to the student.

4.1.3 Response Evaluation

Comparison of a student response to an acceptable answer, determination of whether it is equivalent or not, and provision for feedback (knowledge of results) to the student is hereafter referred to as response evaluation. The feedback can be omitted entirely, can in substance be a simple "yes" or "no", or can range to intricate and lengthy comments (for "correct" responses) or diagnostics (for "wrong" responses). According to Ausubel (3, p. 302): "Subjects who are told why their answers are right or wrong learn more effectively than subjects who merely continue responding and receiving (simple) feedback until they obtain the correct answer." Gronlund (14, p. 365) states that feedback type is important in motivation also. Experiments (2) have shown that students who receive immediate knowledge of correct response after every question perform better on a criterion test than students who receive no such feedback.

Programs written in the TUTOR language normally cause an "OK" or a "NO" to be automatically displayed aside the student response. (This feature can be suppressed by the author.) Additional feedback was to be provided for in each lesson developed.

4.2 Teaching Programs

4.2.1 Kirchhoff's Laws

An investigation of the hierarchal structure of network theory shows that Kirchhoff's Current Law (KCL) and Voltage Law (KVL) are of fundamental importance. Further, each of these laws when first introduced needs a minimal set of prerequisites. These prerequisites and the laws themselves appeared to be easily taught via a computer-based system, thus the first teaching programs decided upon were for KCL and KVL. The programs were to be mainly tutorial in nature, with built-in drill-and-practice. They were to include teaching all of the identified prerequisite skills and facts required for the two laws.

4.2.2 Simple Networks

Ohm's Law and simple linear resistive networks were the next course topics that appeared fruitful to program. Included topics were equivalent circuits, voltage and current division, linearity, and superposition. Tutorial, drill and practice, and test material was needed.

4.2.3 On-line Network Analyzer

One basic philosophy of this research was that of allowing the student to use PLATO in an on-line interactive mode for analyzing networks. Several uses of such a capability were foreseen.

4.2.3.1 What if . . .?

Many times in a learning situation the student will ask the above question. If the available procedures for obtaining an answer are too difficult, or if they take too long, he loses interest. Providing the student with a relatively effortless way to obtain answers facilitates learning, but better yet, it also encourages self-directed study and independence. Model making, simulation and design are also facilitated by an easily used analysis capability. This use is sometimes called the predict-verify approach. It is part of the inquiry mode of learning.

4.2.3.2 Computation

Pure computation of solutions to networks is an obvious use; but of questionable educational value, unless used properly. Embedding the computation in an interactive problem solving process is one valid approach.

4.2.3.3 Laboratory Simulation

The laboratory is probably necessary for the learning of technique. Further, its use for concept learning, proposition learning and discovery learning has been proven to be effective, even though inefficient.* However, laboratory simulation has been shown to be both effective and efficient. According to Entwisle and Huggins (12, p. 386): "(The) greatest contribution (of laboratory simulation) is in laying bare and emphasizing some of the conceptual properties . . . and freeing them from the distraction in which they are ordinarily embedded."

The instructional mode proposed was to introduce most of the theory via programmed text and in lecture-discussion sessions. Then the student would use the on-line analyzer in one or more of the above 3 modes. The student had to be able to easily and quickly enter networks, get their solutions on-line, and then easily modify them for new solutions. The

---

*One frontal attack on inefficiency is that of J.P. Neal and D. V. Meller (21). Their unique approach uses direct sensing of a student'a laboratory activities (such as his instrument dial adjustments and the terminal interconnections he makes) integrated with a computer-based education system.

program was to have a definite small limit on network size, and was to have educational features built-in, such as withholding (at the option of the instructor) computed solutions until pertinent questions could be asked about them. Likewise, questions about elements of the nodal conductance matrix (when treated nodal analysis) or about certain network properties, such as the number of independent meshes, were to be allowed. The program had to compute node voltages, component voltages, and equivalent circuits. It also had to calculate AC response, but would be initially restricted to resistive networks, as was the EE260 course.

4.2.4 Analysis and Simulation of Dynamic Networks

Another use of computational power was decided upon: analysis programs were to be developed for complete solutions to dynamic networks. The student was to be introduced to the theory without PLATO; then he would use the programs for analysis and simulation.

4.2.5 Fourier Sums

A short program on Fourier sums was to be developed, not to teach Fourier theory, but to motivate the student for studying sinusoidal signals.

5. EXISTING LESSONS AND SELECTED RESULTS

5.1 Utility Routines

5.1.1 Drawing of Networks (DRNET)

The basis of the network drawing utility routine, henceforth referred to as DRNET, is the grid of 15 dots in Figure 4. Each dot is a valid terminal for a branch or wire. Branches or wires may be drawn between any pair of terminals that are vertically, horizontally, or diagonally adjacent. The spacing of the dots was chosen after consideration of the PLATO III display size, the size of the branch symbols, the quantity and size of other network symbols (such as voltage and current symbols), required network complexity, and readability.

Display symbols for electrical elements were not available, so these had to be designed as special characters. The minimum size for good appearance and recognition was used. Resistor (R), capacitor (C), and inductor (L) characters (Figure 4), a "black box" character (Figure 17) and an independent source character (Figure 5) were designed. PLATO III does not rotate characters, so a horizontal, a vertical, and two diagonal characters had to be designed for each element.

The grid spacing of Figure 4 appears to be generous, but Figure 5 indicates that when diagonal elements are included, sources are added, and a complement of other network symbols are displayed, the grid spacing may yield marginal clarity. The spacing of Figure 4 was selected since it was considered adequate for the complexity of networks expected. (It proved

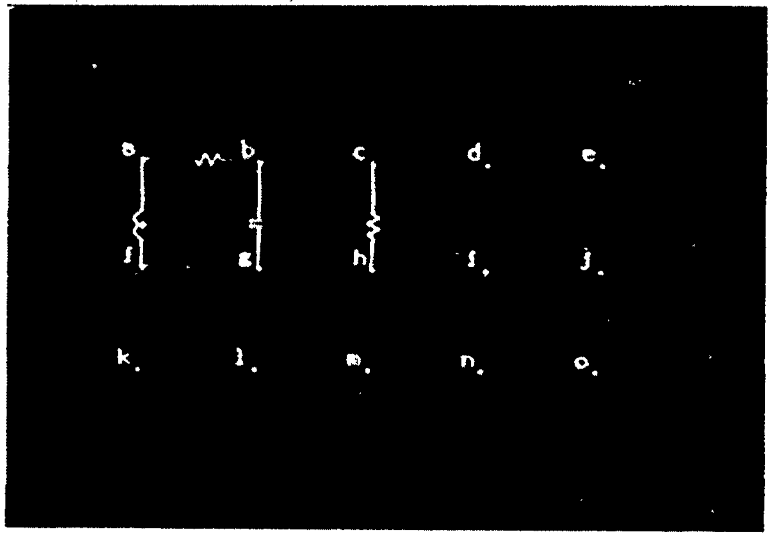


Figure 4. Grid and Network.

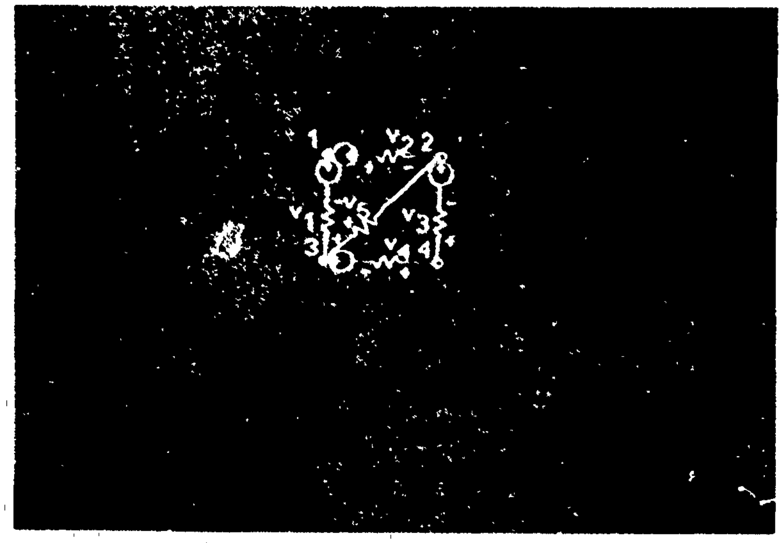


Figure 5. Crowded Symbols.

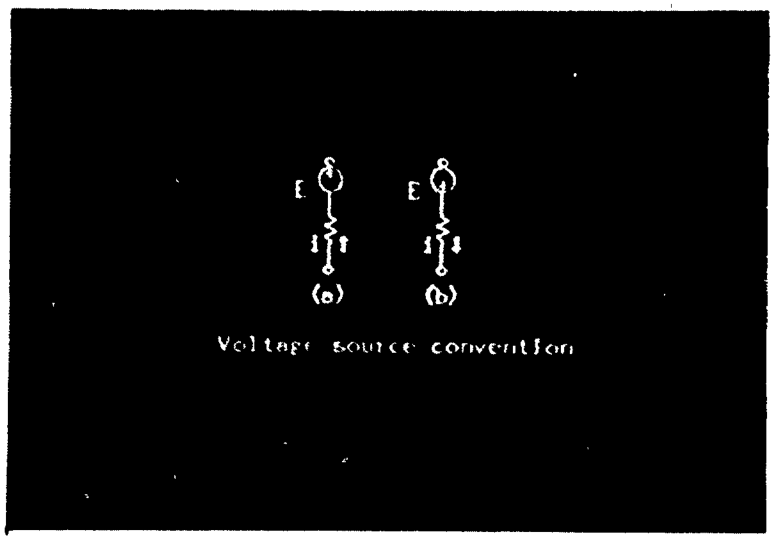


Figure 6. Voltage Source Convention.

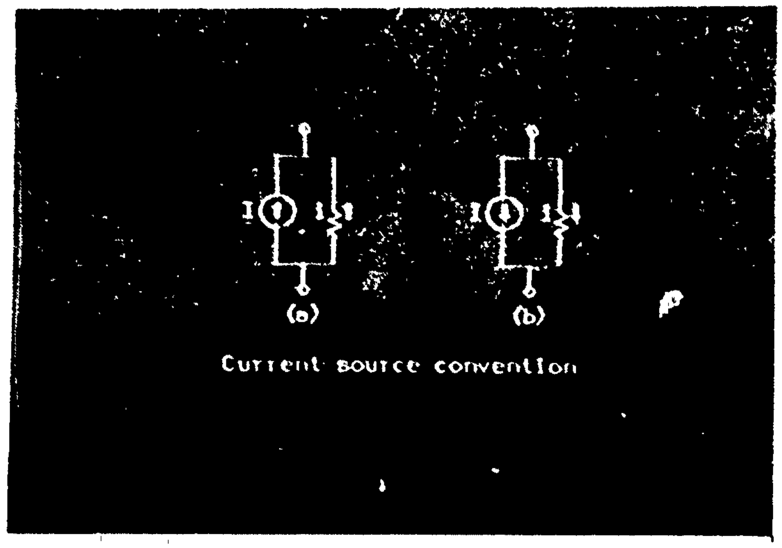


Figure 7. Current Source Convention.

to be adequate in use, also.)

Consideration of both the required network complexity for an introductory network theory course and the PLATO arrangement for program variables resulted in the decision to program DRNET so that 8 branches and 8 wires could be drawn on the grid of Figure 4. However, more complex networks can be built up by "calling" DRNET repeatedly. The limitation for maximum network complexity is the display size, which allows about 50 branches for the chosen grid size.

The specifications needed for each branch are element type (R,C or L), accompaniment (or not) of an independent ideal voltage or current source, two grid terminals, element value, and source value. The grid terminals are stored as a "from" terminal and a "to" terminal to establish references. The reference convention used is that of the Electronic Circuit Analysis Program (ECAP). (This convention is that the current reference arrow points from the "from" terminal to the "to" terminal.) Each branch is allowed either one independent current source or one independent voltage source. The convention used for independent voltage source reference, Figure 6, is that of ECAP. However, the convention used for independent current source reference, Figure 7, is opposite to ECAP. The basis of this choice for DRNET is that Figures 6 and 7 are directly compatible with Thevenin and Norton equivalents, whereas the ECAP convention requires a sign change when converting from one equivalent to another.

Required in addition to the branch specifications are the grid location of the nodes, provision for wires, provision for a "ground" symbol, and the capability to reverse the reference orientation of each branch individually. The display position is determined by specifying the row (1 of 18) and column (1 of 48) for dot "a" of the grid.

For teaching purposes, it was considered desirable to be able to show a given network with or without various network symbols. For example, in Figure 8, node information was suppressed until the student entered a value for n_t, the total number of nodes. After he answered correctly, the node locations and numbers were shown, as in Figure 9. Likewise, after the student entered a value for b, the number of branches, the component voltage and current symbols of each branch were shown, as in Figure 10. The symbols and options that are available appear in Table 1.

When a fixed network was to be presented more than once to a student, the branch, node and wire specifications were programmed into a TUTOR language routine, called a UNIT. The first time the network was to be shown, this UNIT was "called", the options in Table 1 were selected, a display position was specified and DRNET was "called" to display the network and symbols. Each time the same network was used successively without changes, only DRNET needed to be "called".

5.1.2 Ordinary Calculations (CALC)

The TUTOR language has a calculation capability that is easy to use, so the main job of a lesson author is to construct a routine for the student-computer interaction he

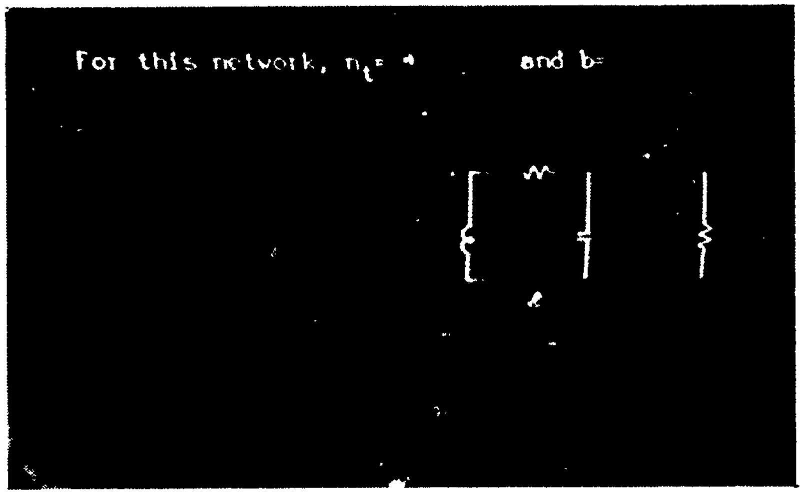


Figure 8. Without Extra Symbols.

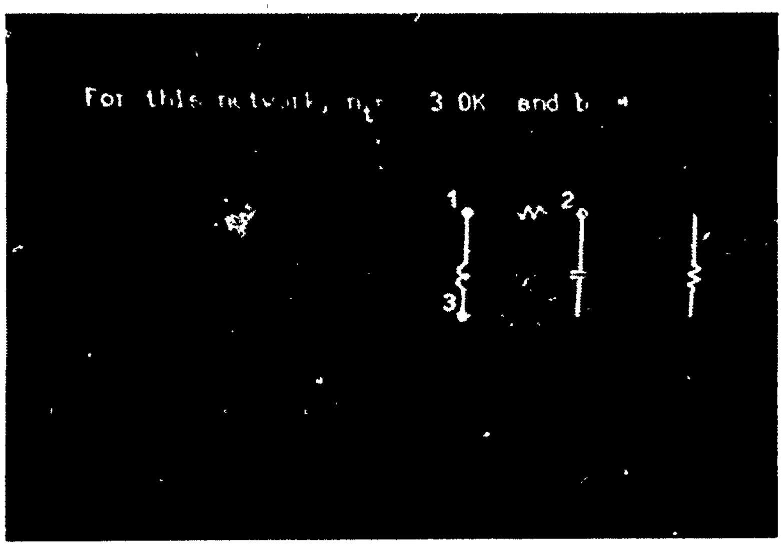


Figure 9. With Node Numbers.

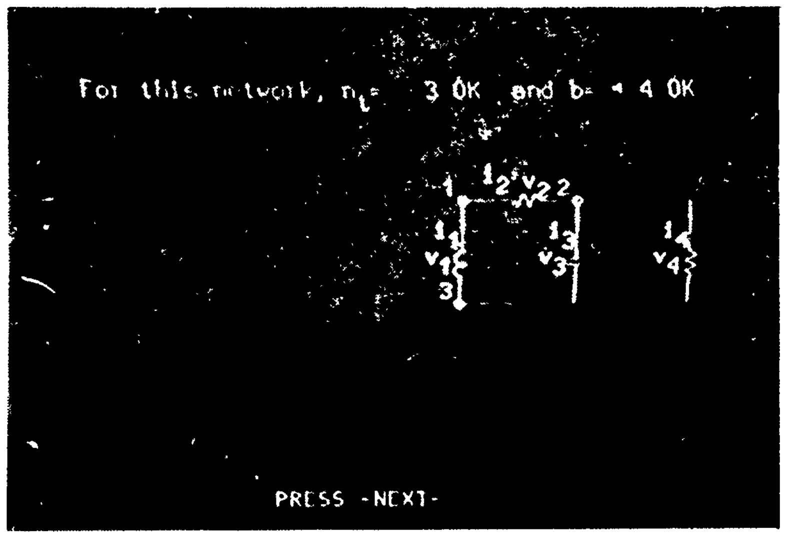


Figure 10. With Node and Component.
Symbols.

SYMBOLS		OPTIONS
Component symbols	1	None
	2	Voltage (v)
	3	Current (i)
	4	Current (j)
	5	Voltage and current (v,i)
	6	Voltage and current (v,j)
	7	Numbers only
	8	Branch (b)
	9	Resistor (R)
Component references	1	None
	2	Voltage
	3	Current
	4	Voltage and current
Node labels	1	None
	2	Node circles
	3	Numbers and circles
	4	Voltage symbols (e), plus numbers and circles
Loop labels	1	None
	2	Loops and symbols (i)
	3	Loops, symbols (i) and references

Table 1. Symbols and Options for DRNET.

desires. Almost the simplest such routine is UNIT CALC, for which the complete TUTOR program is that shown in Figure 11, and for which the display (including student response) appears as in Figure 12. Line 2 of the program of Figure 11 gives the student access to this routine any time he requests the term "calc". (A term is requested at any time in the lesson by pressing key "TERM", typing in the term desired, and then pressing key "NEXT".) Line 3 specifies the 5th line (of 18) and the 10th column (of 48) of the student display for the "WRITE" command of line 4. Line 5 enables student input and causes an arrow to appear at display location 810, as in Figure 12. A display arrow indicates to the student that a response is expected. Line 6 causes the evaluation of a valid numerical expression to be stored in variable F1. Line 7 allows any valid expression to be accepted. During execution, line 10 shows the student the value of his expression.

CALC, even though it uses only 20 words of core storage, is a capable calculator routine. The ordinary mathematical operators are available on the keyset (Figure 3) and the student can use the following functions in his expression: sine, cosine, natural log, power of e, square root, and absolute value. In addition, the author can provide specially defined functions to be compiled with the lesson. Figure 12 shows use of CALC as a simple calculator. Figure 13 shows a more complicated application of the same routine. In this case, the student had first typed "f2=10", then "f3=377", and then "f4=1/60". As shown in Figure 13, he then entered an expression

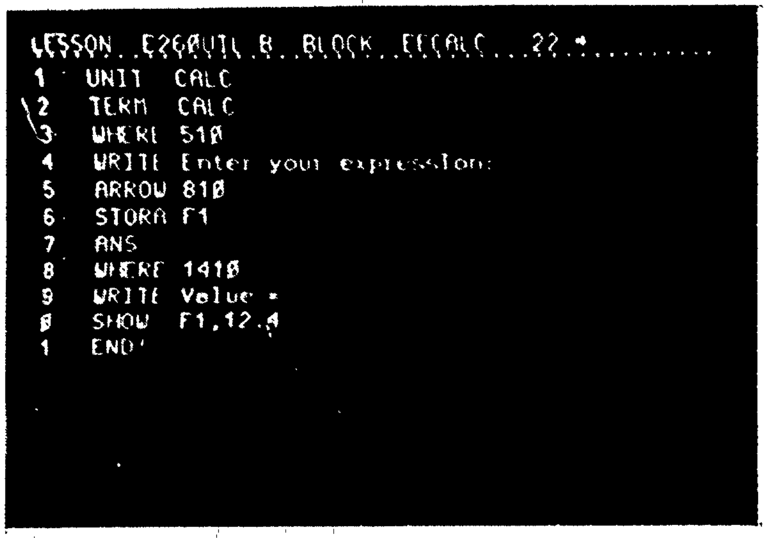


Figure 11. UNIT CALC Program.

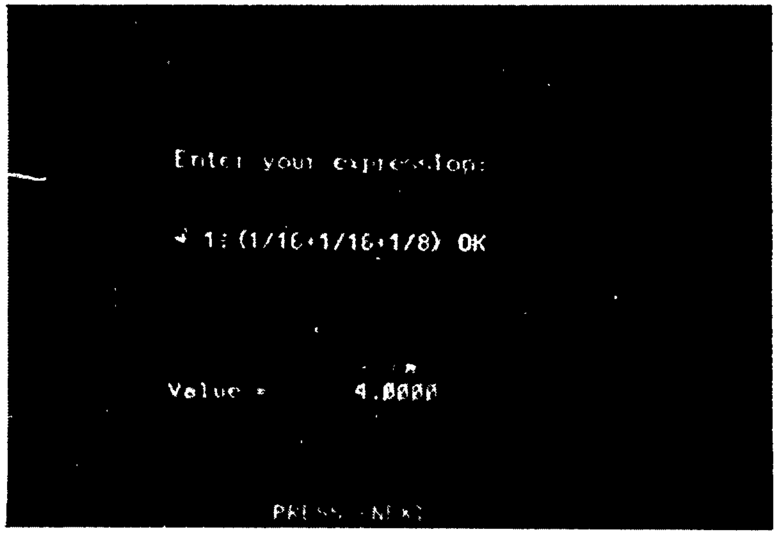


Figure 12. CALC Display.

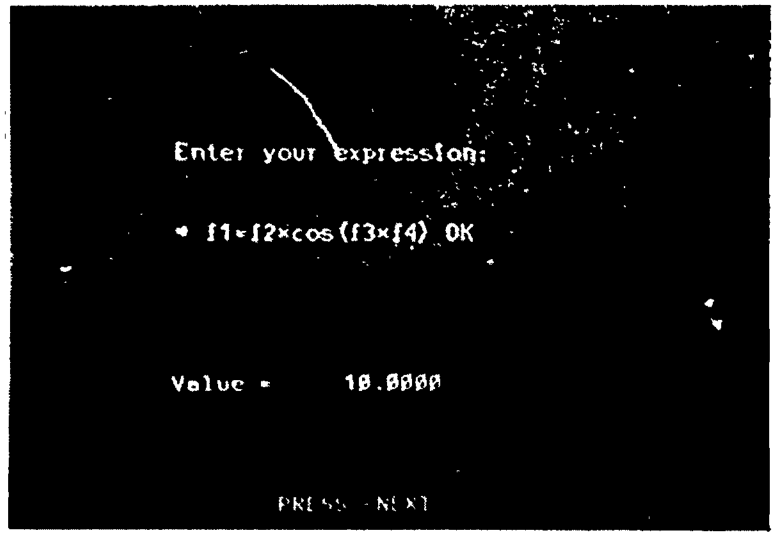


Figure 13. Extended Use of CALC

involving his previously defined variables. The calculation routine was used extensively by the students.

5.1.3 Complex Number Operations

The TUTOR language does not allow for the definition of complex numbers, so a short routine was developed to allow the student to perform elementary operations with them. In the last part of the course this routine was available to the student when he requested the term "complex". The operations available were those in Figure 14. For operations numbered 3 to 6 either complex number could be in either form: polar or rectangular. In Figure 15, the student had previously selected operation 6: Division, and was entering the form of the numbers. In Figure 16, he had entered A in polar form, B in rectangular form, and was then shown C=A÷B in both rectangular and polar form. In addition, he was shown polar plots of normalized nagnitude versus angle for A, B, and C. When entering numbers in polar form, the student can use either degrees or radians for the angle. This routine was used by the students for Sinusoidal Steady State Theory.

5.1.4 Student Comments

The PLATO system has a built in capability for collecting student comments, hence a lesson author merely enables it and provides any interaction he desires. The student's comments are recorded and annotated with his name, the time, and his place in the lesson. The data is sorted, printed, and given to the lesson author the morning after the PLATO session. The student can comment any time he requests

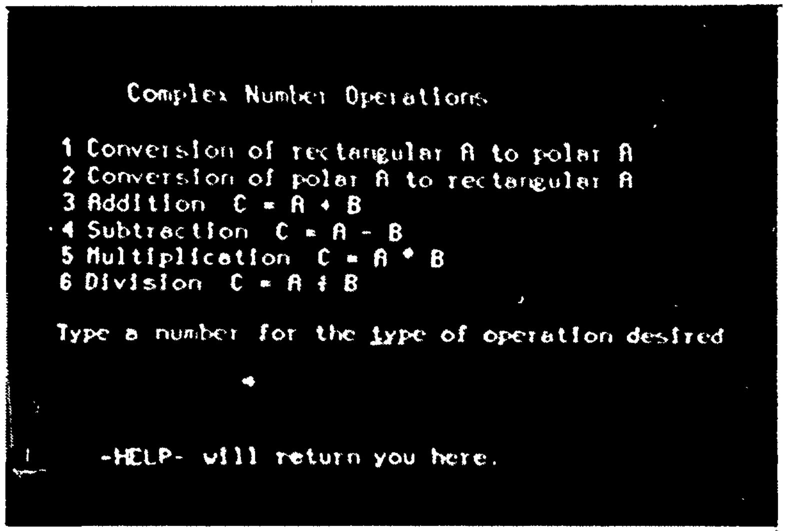


Figure 14. Complex Number Operations.

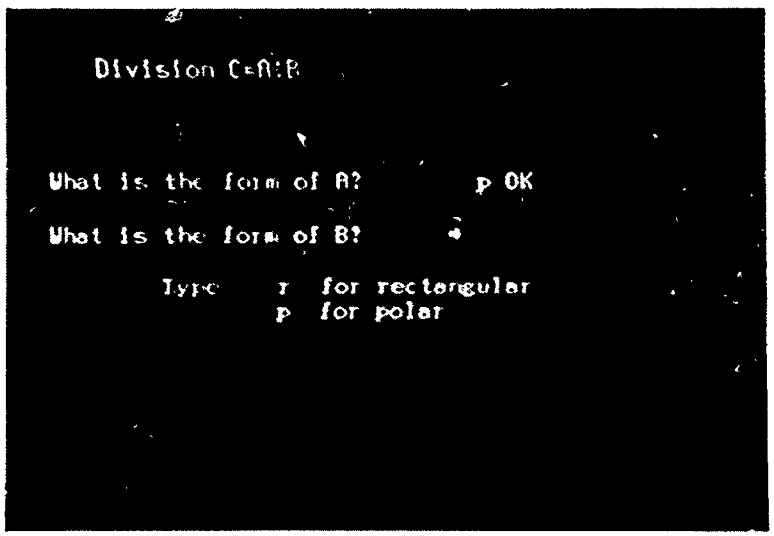


Figure 15. Form Specification.

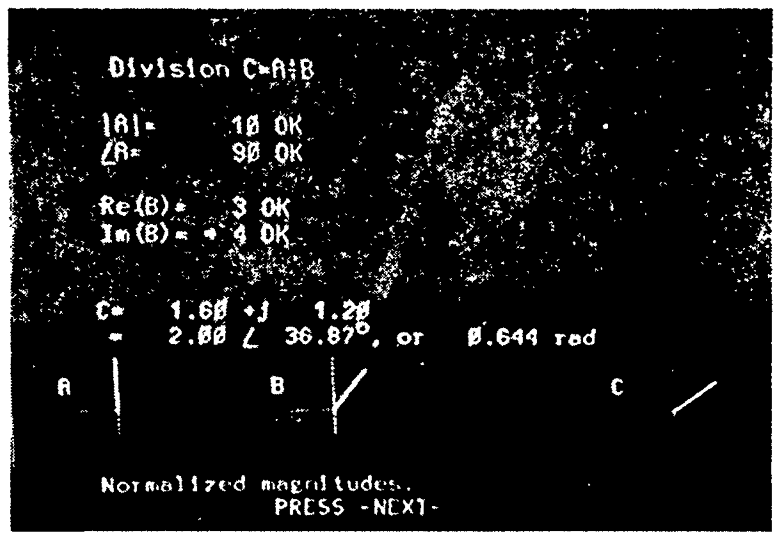


Figure 16. Complex Number Evaluation.

the term "comment". The students used this capability quite freely. Many comments were useful for revision of the lessons.

5.1.5 Response Evaluation

Response evaluation routines were developed primarily for each teaching program rather than for the course as a whole.* Examples of response evaluation from several different lessons are discussed here collectively. The automatic "OK" or "NO" provided by the TUTOR language can be seen in Figure 17 through 21.

For Figure 17, the correct Kirchhoff current law (KCL) equation is $i_1+i_3-i_4=0$. The response evaluation routine accepted any algebraic equivalent of this equation. Since voltage symbols were intentionally shown on the network instead of current symbols (see Section 5.22), the error of Figure 17 was anticipated and the diagnostic "Try KCL" was given if the student used "v" symbols instead of "i" symbols. For Figure 18, the correct answer is e_2g_3 or g_3e_2. Again, the error was anticipated since previously the students had expressed currents only in terms of the "v" and "r" symbols.

The response evaluation routines disallowed identities, such as in Figure 19. Simultaneously, they allowed equivalent algebraic expressions. For Figure 20 the correct answer is $i_1+i_3-i_4=0$. The routine correctly interpreted sin $(0)i_2$ as zero and $0i_5$as zero; it also interpreted i3 as i_3. It then decided

---

*Bruce A. Sherwood (22) has developed a TUTOR language routine that handles general algebraic expressions and equations. Some of the routines discussed here were based on his work.

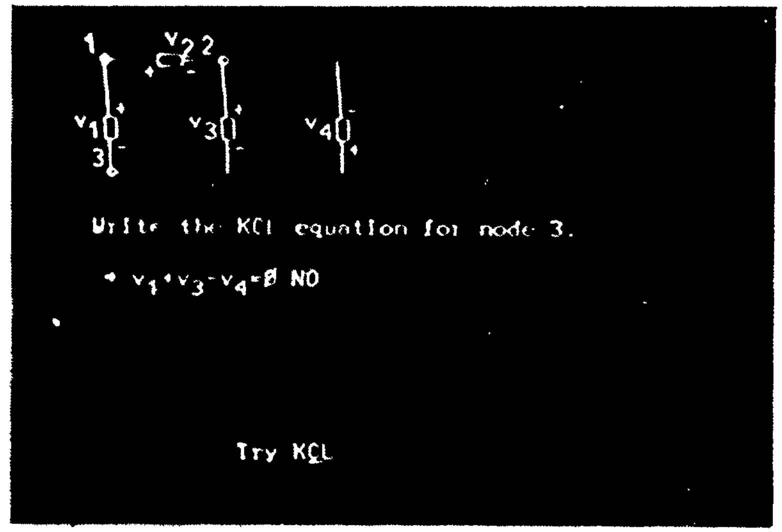


Figure 17. Current Law Diagnostic.

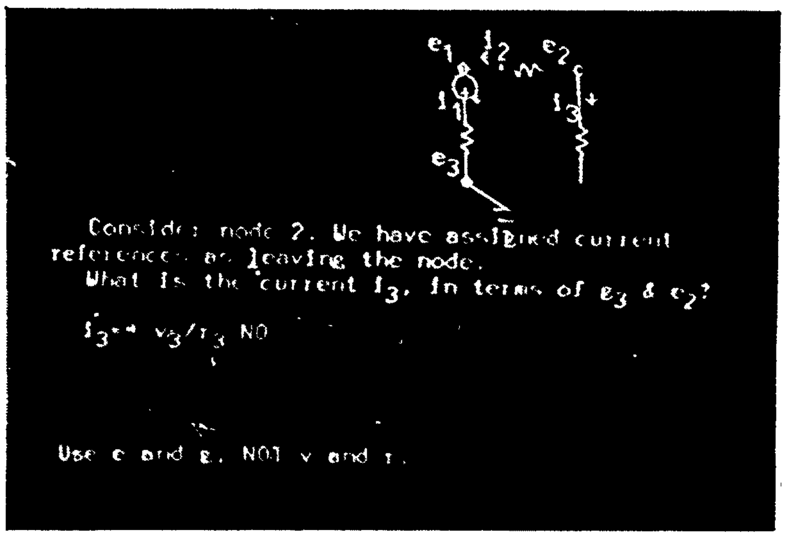


Figure 18. Node Voltage Diagnostic.

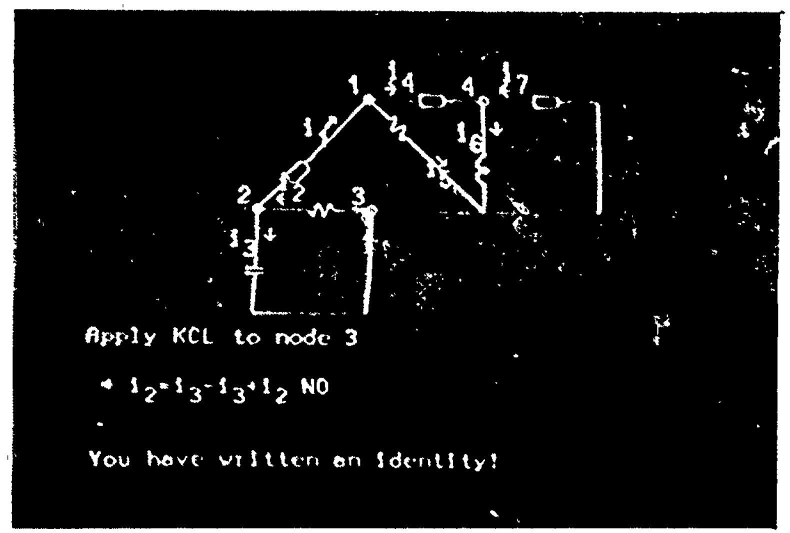


Figure 19. Identity Diagnostic.

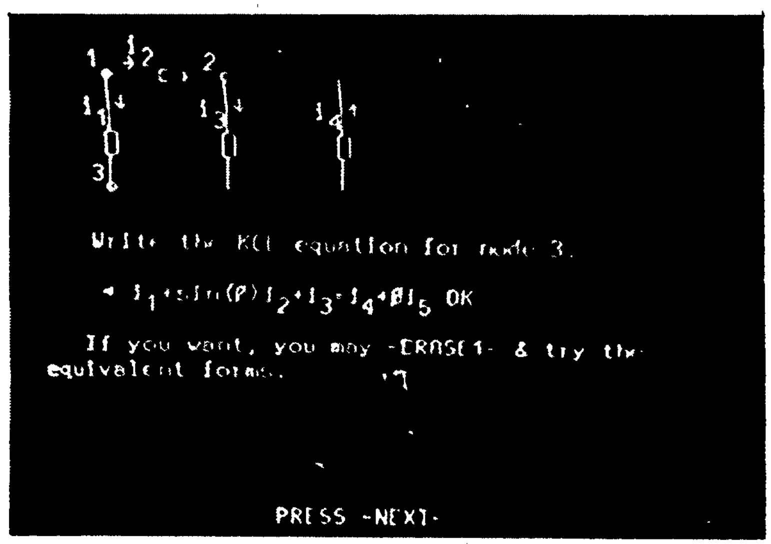


Figure 20. Acceptable KCL Equation.

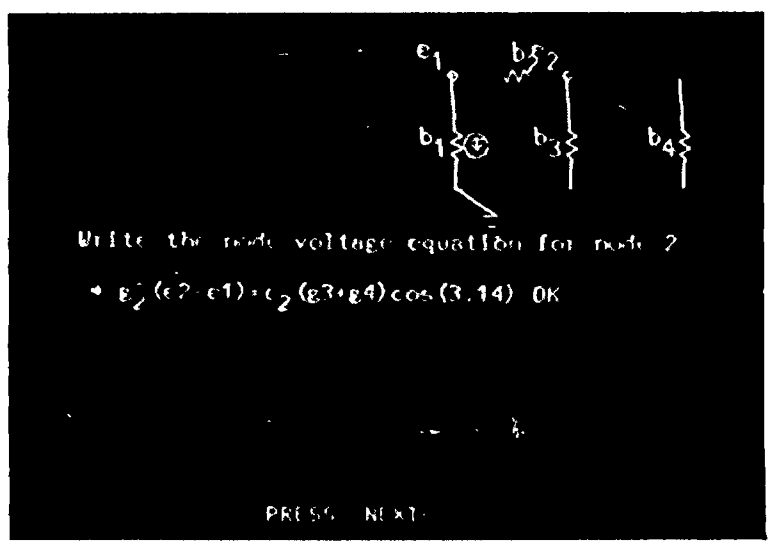


Figure 21. Acceptable Node Voltage Equation.

that $i_1+i_3=i_4$ was equivalent to $i_1+i_3-i_4=0$. The student was allowed a free form of expression; he was, for example, allowed to use implied multiplication. Another example is the equation of Figure 21. For other examples of diagnostics see Figures 25 and 26 and Section 5.21.

Many favorable student comments were prompted by the response evaluation routines. One student commented: "Good diagnostics." Another commented: "The equation recognizer is a rather significant device."

5.2 Teaching Programs*

5.2.1 Kirchhoff's Current and Voltage Laws (KCL and KVL)

First the prerequisites for KCL and KVL were introduced. Then KCL was stated (Figure 22), the student was shown a network (Figure 23), and he was allowed to arbitrarily assign his own current references. Keys with arrows on them, Figure 24 and 3, were used for indicating directions. In Figure 23 the student has chosen references for i_1, i_2, and i_3. (The small horizontal arrow by i_4, in the lower right, is a PLATO system arrow which indicates that a student response is expected.)

Next (Figure 25) he was asked to apply KCL at node 2 by giving the values (0, +1, or -1) of the coefficients of the equation shown him. If the student pressed the "HELP" key he was provided with his first example. (Only 4 out of 11 requested help.) Further help was available by using "SHIFT"

---

*Some results from the second semester (Spring 1971 class of 11 students are presented in this section.

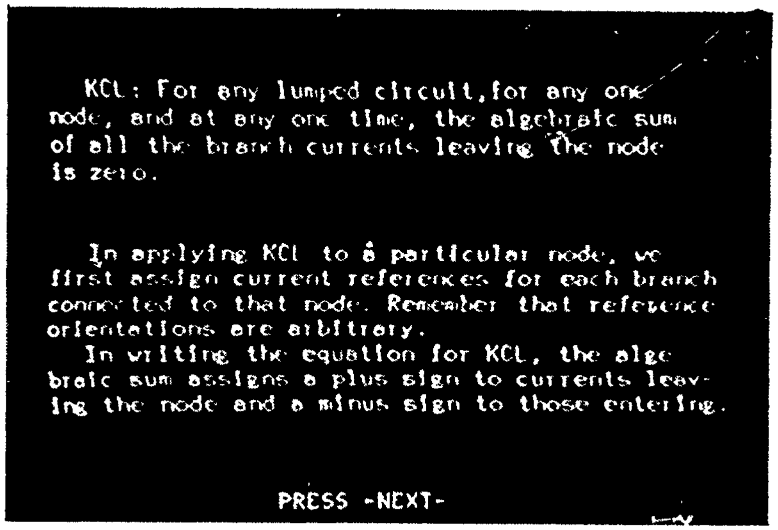
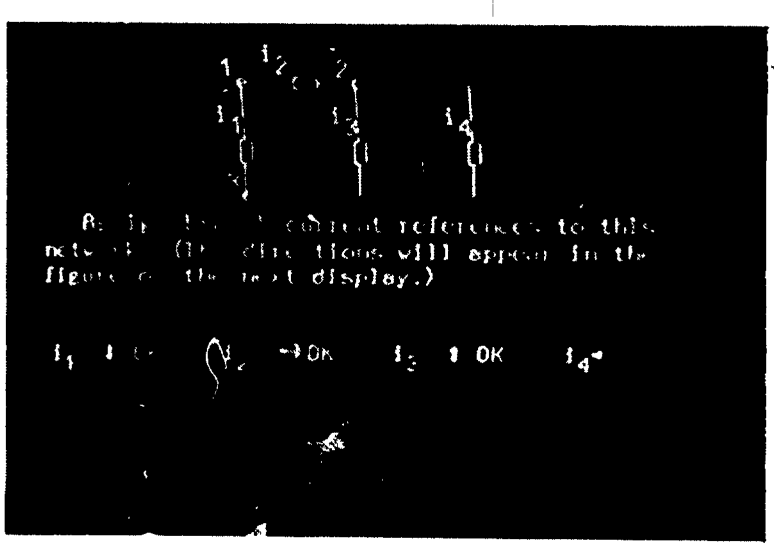


Figure 22. Kirchhoff's Current Law.



F'‸ᵤᵣᵉ 23. A‸signing Current References.

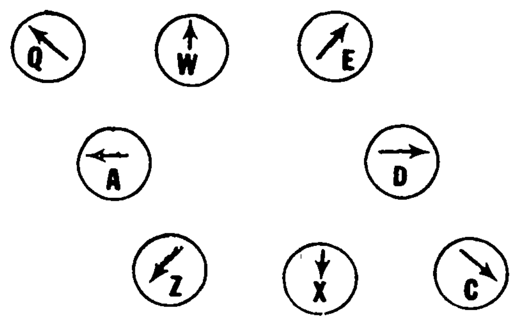


Figure 24. Keys with Arrows.

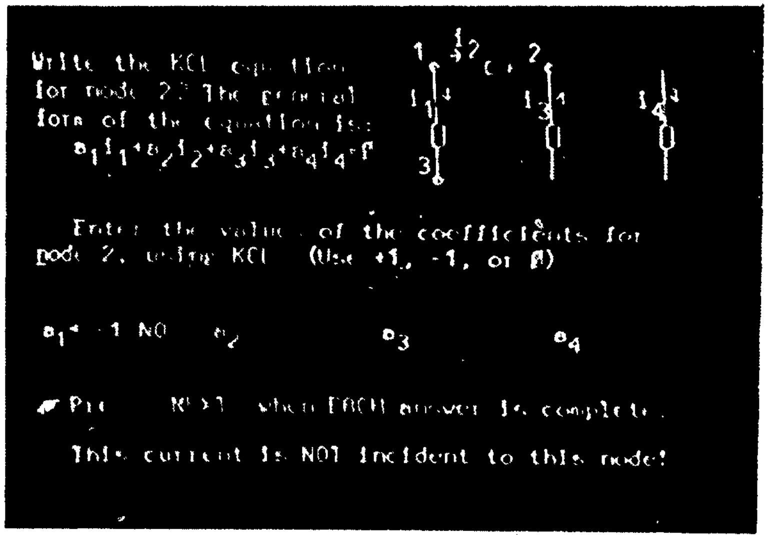


Figure 25. Application of KCL.

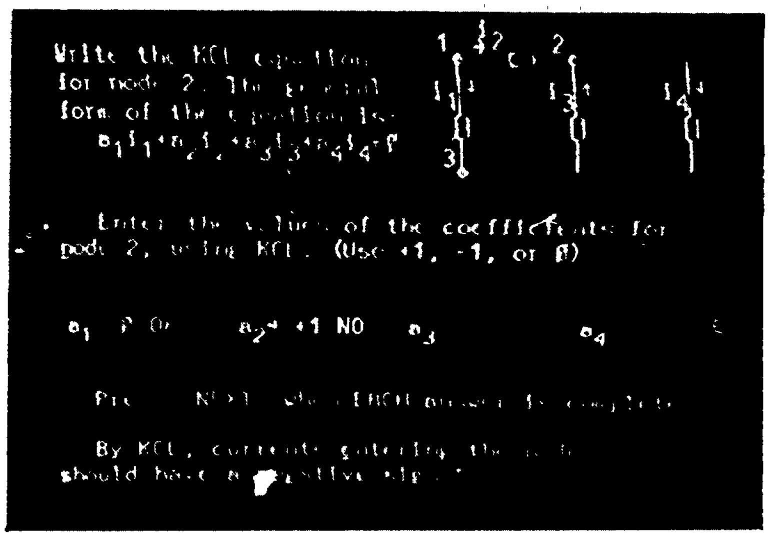


Figure 26. Diagnostic Feedback for KCL.

plus "HELP", referred to as "HELP 1"; in this case, it provided a more basic example. (None of the 11 used "HELP 1".) If the student entered the wrong coefficient, appropriate feedback was given him. For example in Figure 25, the student entered "-1" for the value of coefficient a_1. PLATO "judged" his response "NO" and showed the diagnostic feedback on the last line of the figure. A further example of diagnostic feedback is provided in Figure 26. The student was not allowed to go on until all four coefficients for node 2 were responded to correctly. (If the author desired, a press of key "ANS", (Figure 3), showed the student a correct answer; however, it had been inhibited here.) Finally, the student was asked to apply KCL to node 3 of the same network.

The correct coefficients had to be calculated internally for comparison with the student's responses. The approach used was to set up a node to branch incidence matrix for the network shown to the student. The matrix was modified (when necessary) by the references chosen by the student, thus the correct coefficients were available from the matrix.

If the student made 2 or more errors out of the 8 coefficients, he was forced to go back, assign new references, and repeat the above procedure. (Out of 11, 3 were forced back; all 3 had asked for help.) (Keypresses other than 0, 1, and -1 were ignored.) If he made less than 2 errors, he had the option of going on to new material or of repeating with new references. (Out of 11, 3 repeated; one repeated twice.)

KVL was handled by a similar procedure. A loop was defined, the student assigned arbitrary loop orientations, then he applied KVL to two loops. (For KVL, none were forced back. Also, only one opted to repeat KVL. The average completion time for KCL and KVL was 18 minutes. The range was 13 to 26 minutes.) Other material in this lesson tested and extended concepts of reference orientations. It also introduced the 3 algebraically equivalent forms of KCL (one of which is: currents entering = currents leaving). The concept of associated (or load set) references was introduced just before KVL.

This lesson consisted of a combination of tutorial, drill and practice, and test. The "test" was provided by making exit from the drill and practice contingent on an acceptable error rate. One philosophy of the lesson was to display only the minimum of both tutorial and procedural material. Appropriate additional information was given if the student asked for help or erred. The above discussion about the "HELP" key exemplifies this for tutorial material. An example for procedural material is the use of the "arrow keys" in choosing reference orientations for Figure 23. The students had never used them before nor received any instruction about their use. Yet when confronted with the display of Figure 23, four out of eleven used them properly. Two pressed "HELP", thereby receiving instructions; the remaining 5 pressed a wrong key, which showed the instructions automatically. This lesson was one of the best liked. One student commented: "Best lesson I ever had."

5.2.2 Applications of KCL and KVL

The above application of KCL and KVL to just evaluating the coefficients one at a time was followed by this lesson on more advanced application. First, the student was told that with the associated reference (or load set) convention, if voltage references were given, then the current references were determined, and vice versa. Then the student was shown a network (Figure 27) and asked to write the complete KCL equation for node 3. As the figure shows, the voltage symbols and references were given, but he was asked to apply KCL. Key "HELP" gave the hint in Figure 28. If the student then used "HELP 1", he could ask for the current symbols instead; if so, he next saw the problem in Figure 29. (None of the 11 students pressed "HELP" or used "HELP 1".)

After this the student was given a handout containing 3 networks. Each network had 3 to 4 loops and nodes, the nodes and loops being indicated by numbers on the handout. The program asked the student to select a network, select KCL or KVL, and then select a node or loop. He was then asked to write the appropriate law for his choice of node or loop. Any response algebraically equivalent to the internal author-provided answer was accepted. One student selection, with response, is shown in Figure 30.

This lesson provided drill and practice for application of KCL and KVL to arbitrary lumped networks. (The average completion time was 37 minutes. The range was from 25 to 52 minutes.) The student had to respond correctly to each question

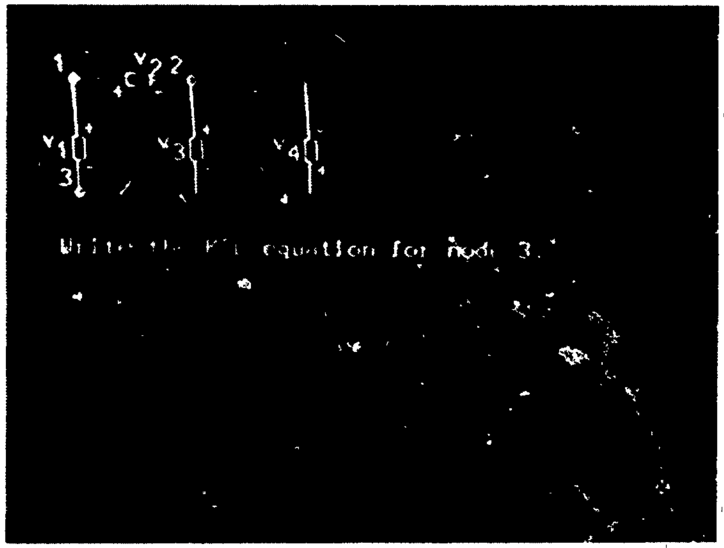


Figure 27. KCL with Voltage References.

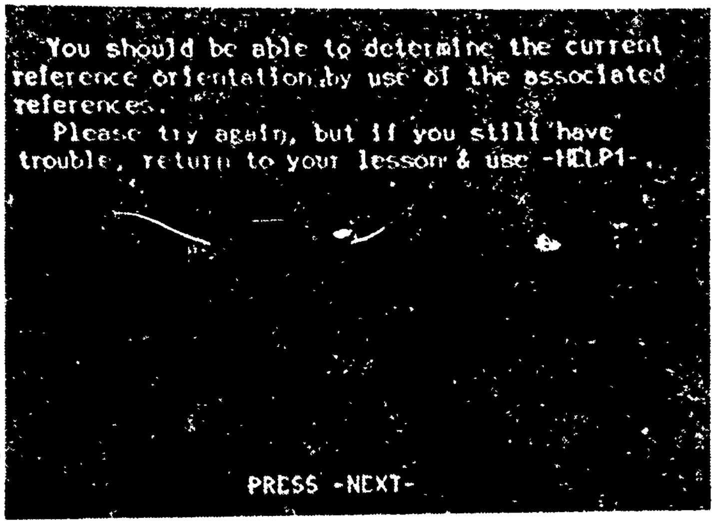


Figure 28. Associated References and KCL.

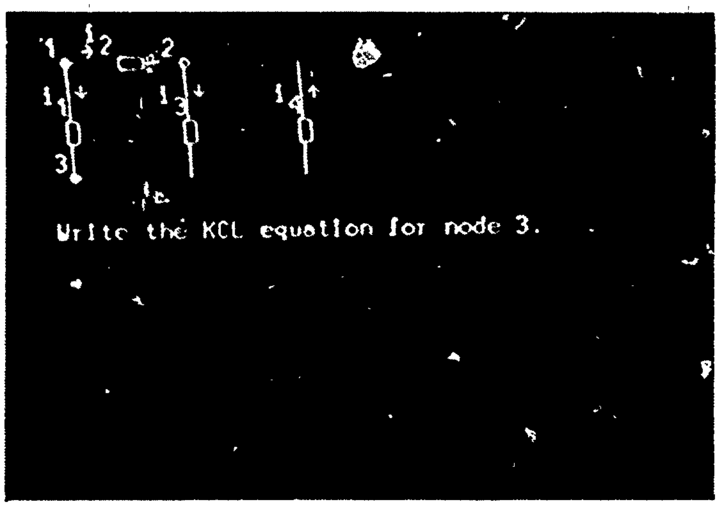


Figure 29. KCL with Current References.

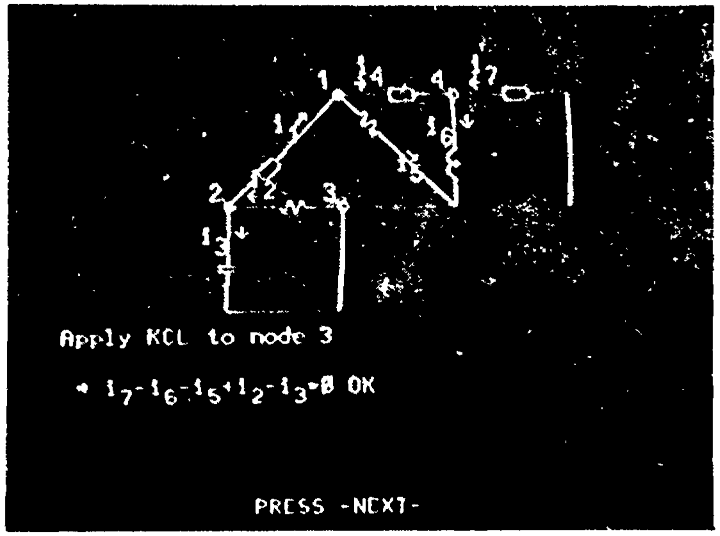


Figure 30. Network for Drill and Practice.

to proceed. He received diagnostics for errors, but not an answer. Reading of the pertinent pages in a programmed text (5) was assigned before the PLATO class. It was assumed that a student who completed this lesson on PLATO could apply KCL and KVL at the level of this lesson (which did not include v-i relations). These topics were never treated in a non-PLATO class. One student commented that he liked the feature of "as much practice as I want, with instant grading so I can decide if I know it or not."

5.2.3 Simple Equivalent Networks

Two lessons were developed on various aspects of simple equivalent networks. Outside reading in a programmed text (5) was assigned to be done before the PLATO class since these lessons did not introduce the topics; rather, their purpose was to test and extend understanding of the various principles and concepts.

The philosophy of the first part was to give the student some freedom of expression in responding and a chance to attack problem solving. The second question of Figure 31 allowed the student to express the v-i relation in his own format; the way that is understandable to him. His expression was then used on succeeding displays. In Figure 32 the student had his first opportunity to attack a multistep problem; the question shown was the first one that hinted at a dialog type interaction. The expected response was something equivalent to "label the axes". "Judging" of the response was by the keywords "label" and "axes" or their synonyms. (Two out of 11 responded successfully.)

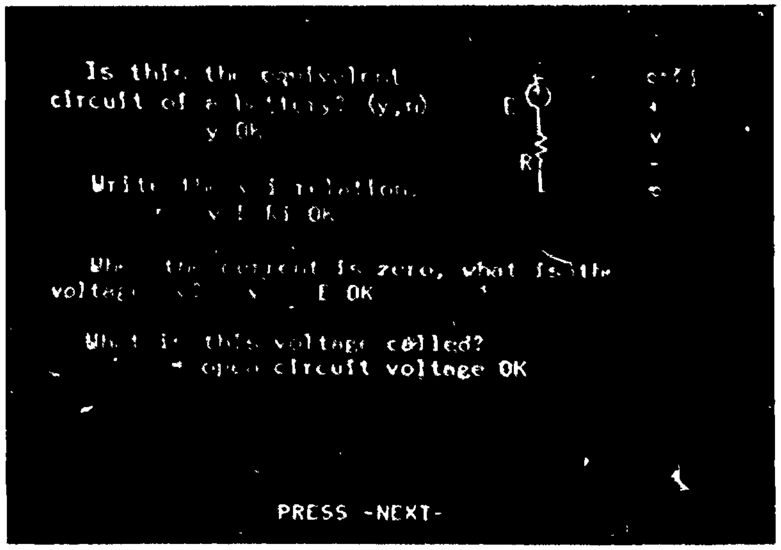


Figure 31. Equivalent Networks.

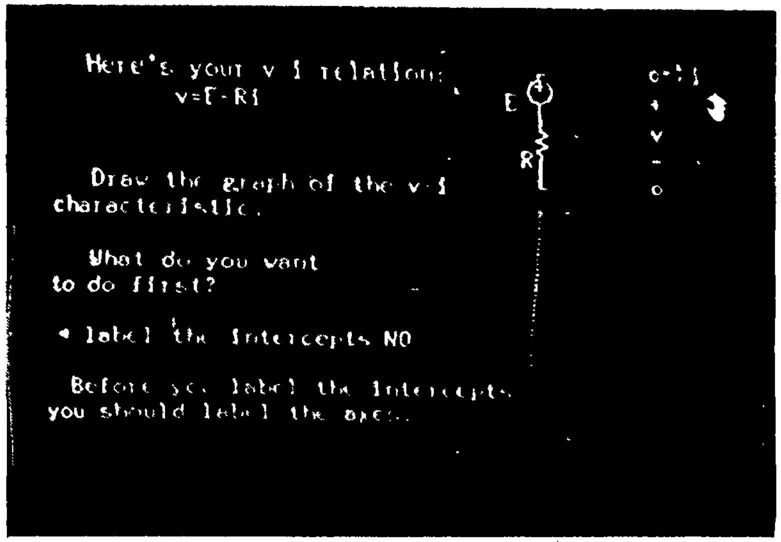


Figure 32. Dialog Attempt.

Figure 32 shows the feedback for one response, Figure 33 for another. Key "HELP" listed some of the words that were in the "vocabulary". (Four out of 11 pressed key "HELP" immediately; five were shown the feedback of Figure 33.) He was free to label the axes as he desired; whatever his choice, it was used in his next display, as in Figure 34. Here he was to define the line by selecting two end points, using key "m" to mark each point, and using the "arrow keys" to move the "+" marker. (All used key "HELP" to find out how to mark the end points.)

The student's graph of Figure 34 had the wrong polarity slope. In Figure 35 he was given the chance to redraw the graph (if further thought revealed his error) or to continue. When his graph was incorrect, as in Figure 35, he was shown what was wrong with it. (only 1 student had the wrong slope. He had first drawn it correctly, then in Figure 35 he opted for redrawing, which he then did incorrectly. He was the only one to opt for redrawing. He may have been testing the program) After drawing it correctly he was asked to give the intercepts (Figure 36), for which diagnostics were provided for incorrect responses.

An important concept in finding the equivalent resistance of a network is to render the independent sources "dead", which means to leave the source in the circuit, but to reduce its magnitude to zero. Students have difficulty in understanding why one should do this. An attempt to show why by a graphical v-i relation approach is shown in Figure 37. The student was first shown the circuit and the graph of its v-i relation (shown

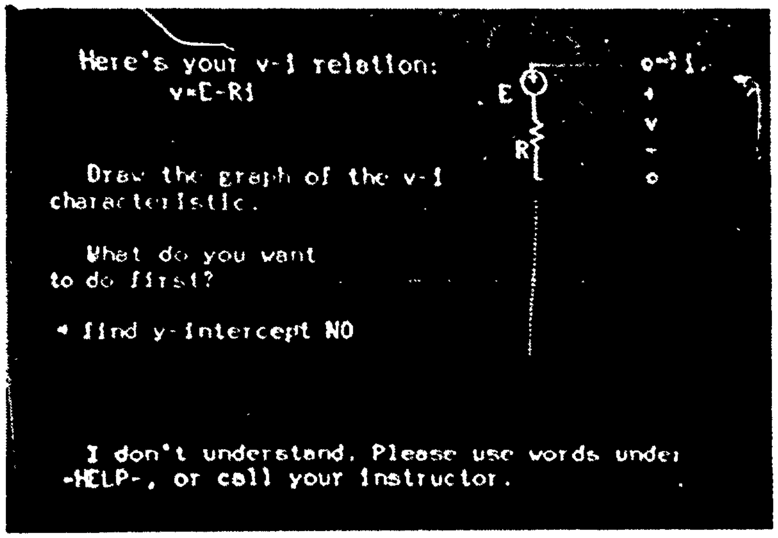


Figure 33. Dialog Failure.

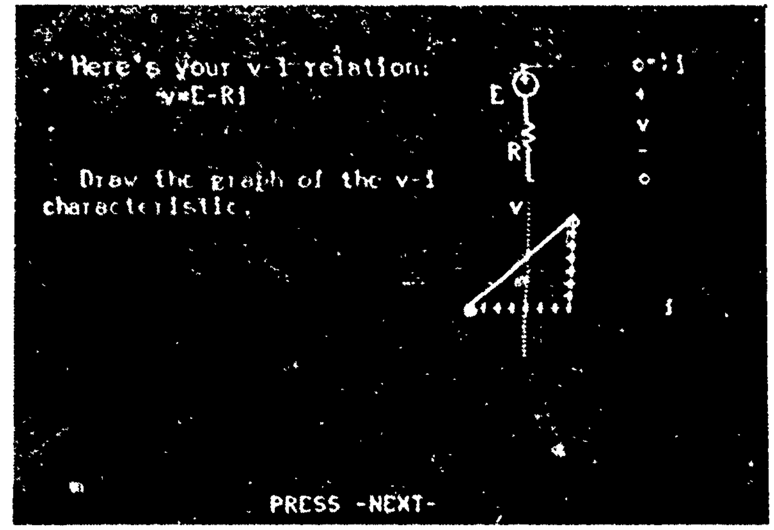


Figure 34. Graphing a v-i Relation.

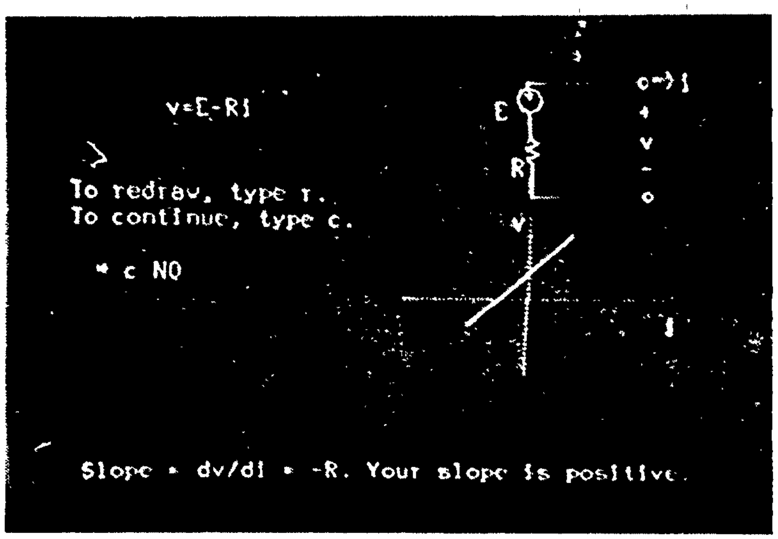


Figure 35. Graphing Diagnostic.

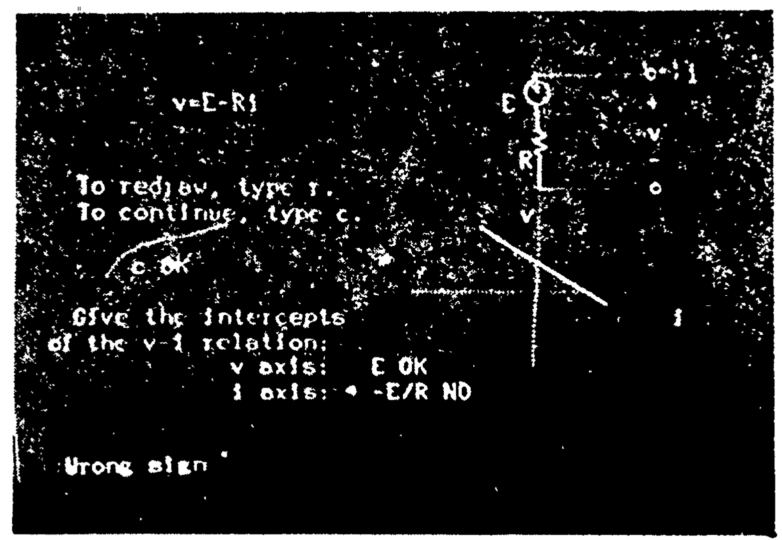


Figure 36. Sign Diagnostic.

with the intercepts E and -E/R). When he "made the source dead" by typing "Ø" (zero), the graph of the new v-i relation (the line through the origin) was drawn. He was then asked what element type was represented by this new graph. (One student responded "ideal source", the other 10 responded "r", "resistor", or "ideal resistor".)

The main topics explored in these two lessons were current and voltage sources, Thevenin and Norton equivalents, series and parallel combinations, voltage and current division, linearity, linear combinations, and power. (The average completion time for one lesson was 25 minutes; the range was 15 to 34 minutes. The average completion time for the other lesson was 37 minutes; the range was 12 minutes to 97 minutes.)

5.2.4 Nodal Analysis

A short lesson was developed as an introduction to nodal analysis. It consisted of tutorial material on reference nodes and node voltages and gave the student practice in writing nodal equations. Two excerpts are shown in Figure 38 and 39. (The average completion time was 26 minutes; the range was 19 to 35 minutes.)

5.2.5 On-line Network Analyzer

A program was developed to allow the student to analyze linear steady state RLC networks on-line in an interactive mode. The maximum size network allowed was 8 branches and 8 nodes. One controlled current source was allowed; it could be either current or voltage controlled. In DC analysis the student could choose nodal analysis or equivalent circuit

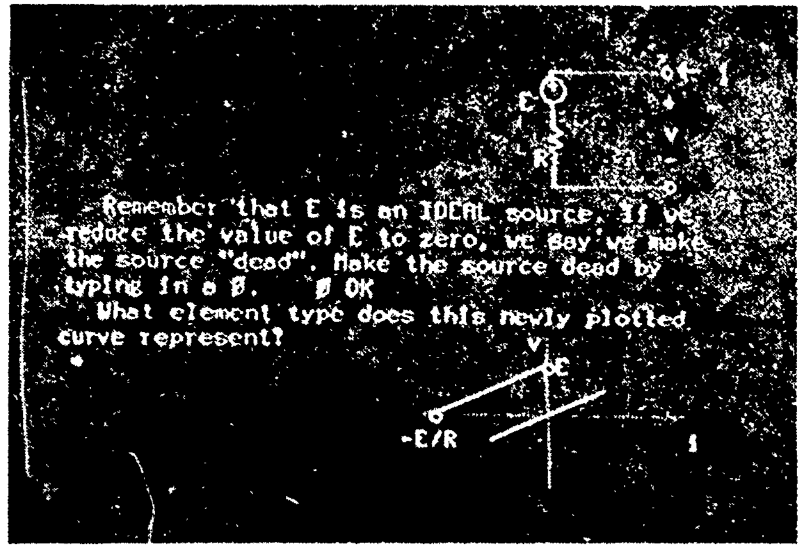


Figure 37. Rendering a Source "Dead".

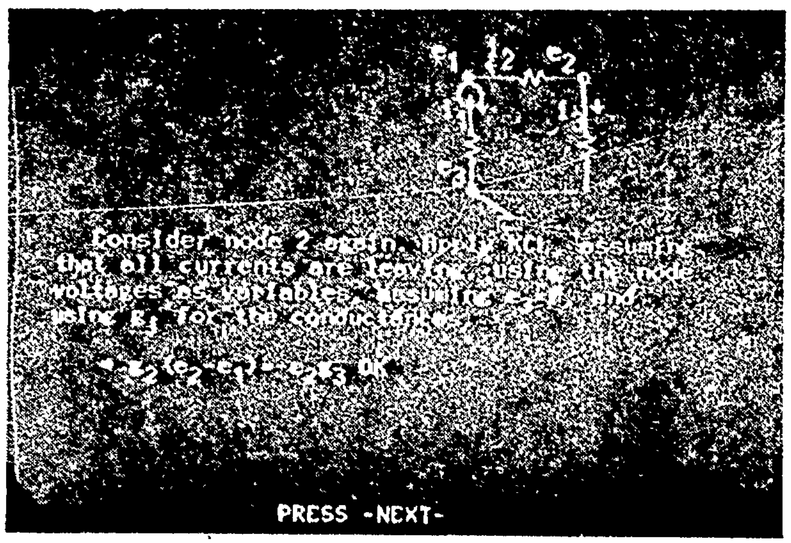


Figure 38. Introduction to Node Voltage Equations.

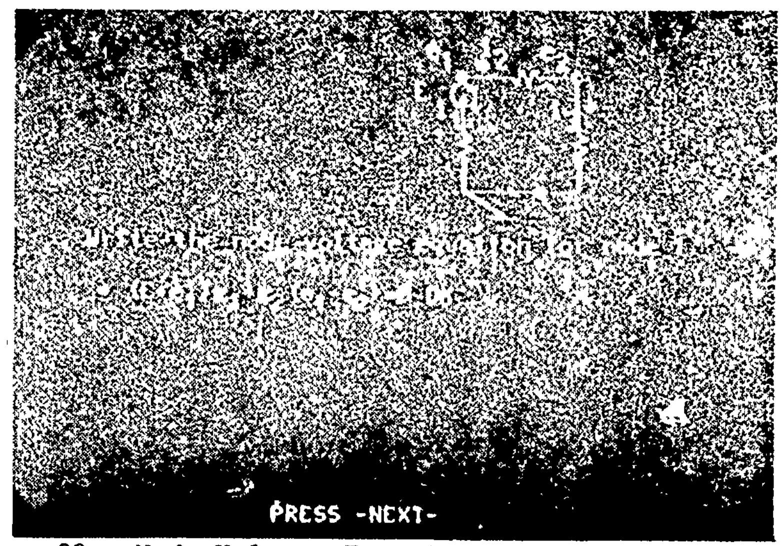


Figure 39. Node Voltage Equations.

analysis. Nodal analysis provided the node to datum voltages, from which the component voltages and currents were obtained. Equivalent circuit analysis calculated the Thevenin voltage and impedance of all nodes with respect to a datum node. In AC analysis, the student could select, in addition, frequency response data. A magnitude versus frequency plot was provided as part of this data.

First the student chose DC or AC analysis. Then he was shown the "entry" display, which appeared as in Figure 40. Some "HELP" key instructions are shown in Figure 41. Each branch was entered one at a time, as indicated in Figure 42, and was then added to the network, as in Figure 43 and 44. The student could inspect the branch values at any time by pressing key "DATA", which showed the information in Figure 45. He could modify the values of an individual branch (from the Figure 40 display), could delete branches and wires, as in Figure 46, or he could erase the complete network at any time.

After the student indicated his network was complete, two calculations had to be made because of the allowance of wires. First the nodes had to be identified, and second, a check was made for shorted branches. In addition, a check was made for the proper specification of the controlled source, if there was one. The program at this point had a built-in branch that could be used or bypassed at the option of the instructor. If used, it randomly selected one of 5 questions, one of which is shown in Figure 47.

Next, Figure 48, the student could select the type of

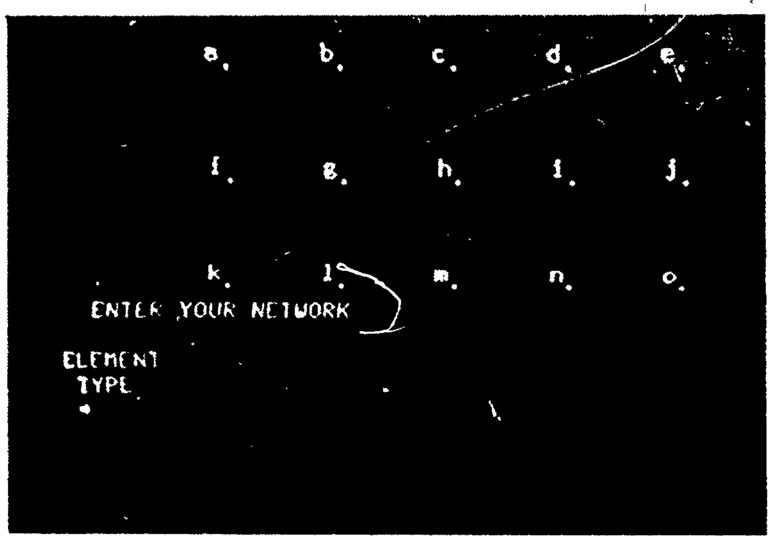


Figure 40. "Entry" Display.

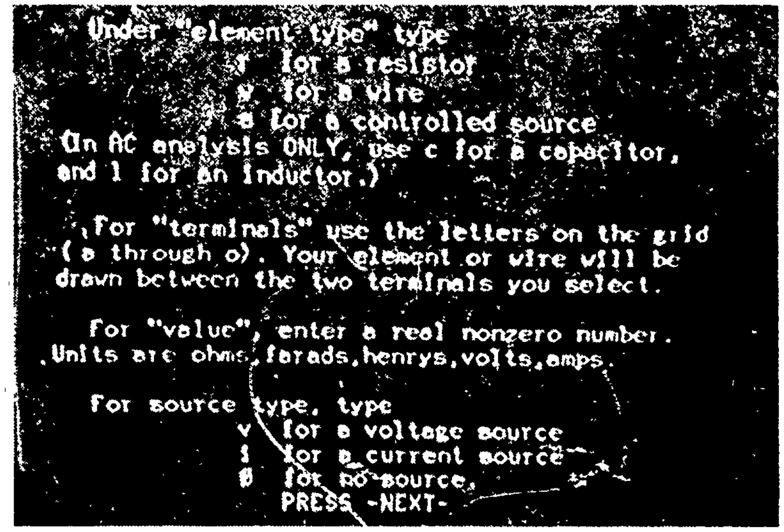


Figure 41. Entry Help.

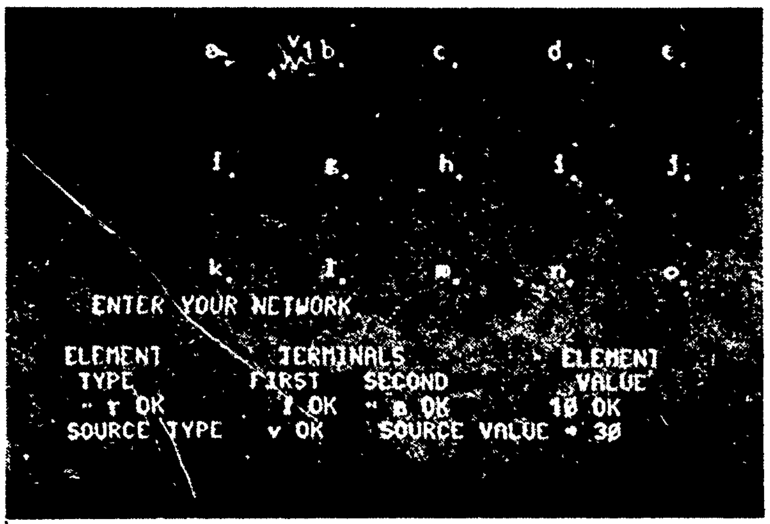


Figure 42. Branch Specification.

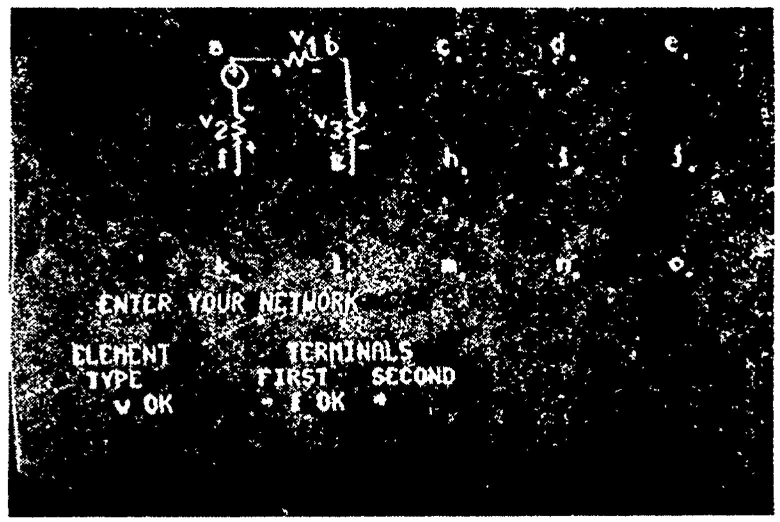


Figure 43. Wire Specification.

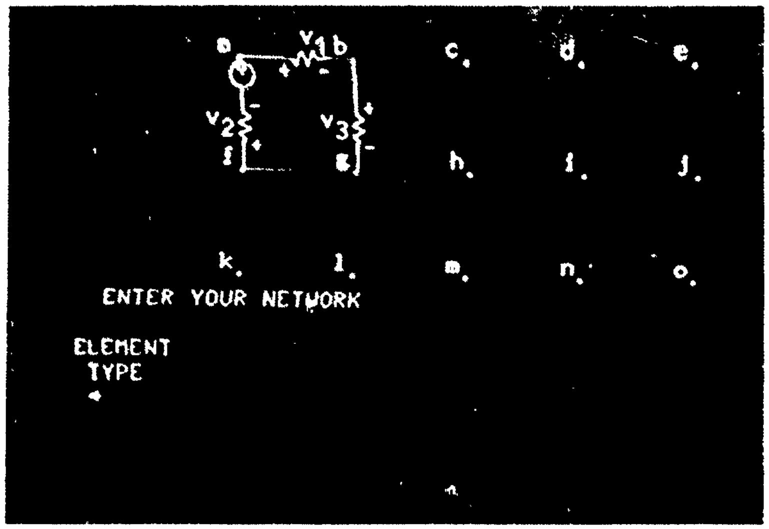


Figure 44. Complete Network.

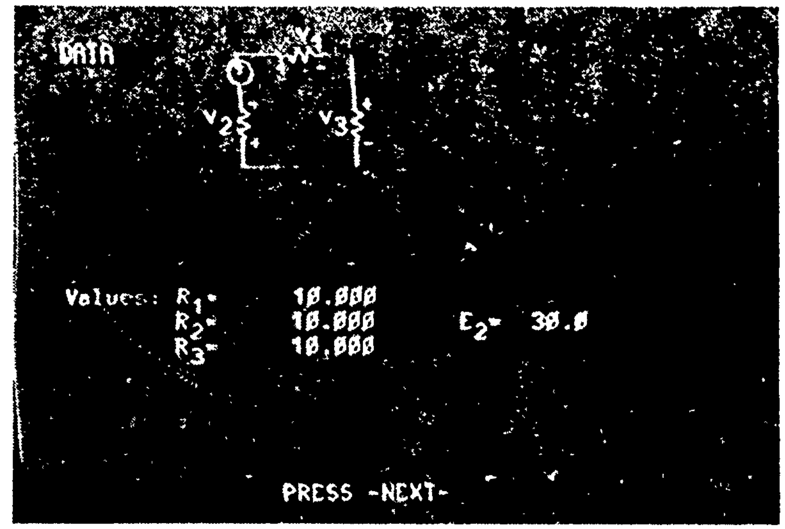


Figure 45. Network Data.

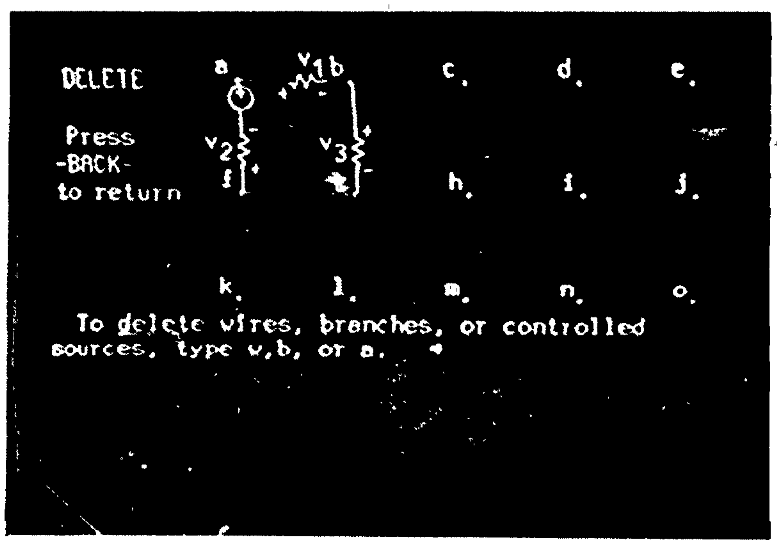


Figure 46. Delete Capability.

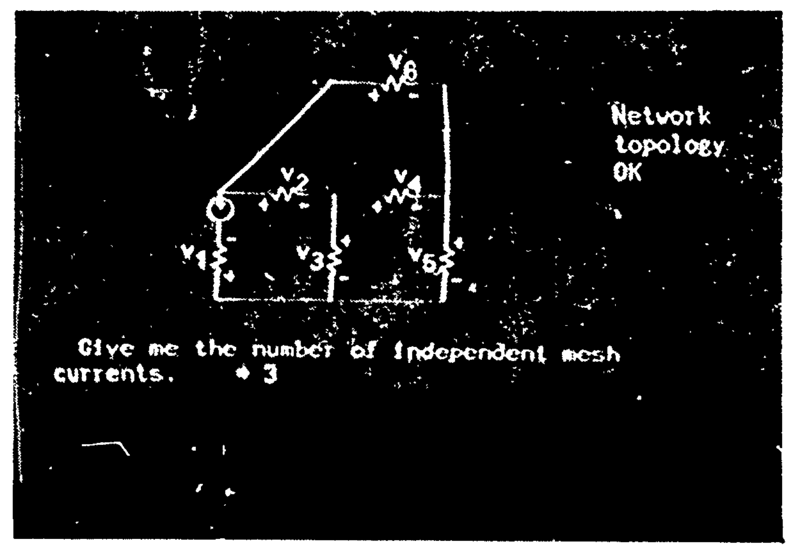


Figure 47. Random Question.

analysis and the reference, or datum, node. The node voltage solutions were shown as in Figure 49; the equivalent circuit solutions were as in Figure 50. In AC analysis, frequency response data could be selected. After entering his network, such as the one in Figure 51, the student could select frequency response, enter his frequency data as in Figure 52, then make a choice of one of the plots of Figure 53. If he selected "voltage gain" (choices 3 or 4) he could specify the nodes, as in Figure 54. Two "plots" are shown in Figures 55 and 56. Only 4 data points (frequencies) were calculated at a time. This was done for two reasons. First, this method lowered the peak computational load on the computer at any one time. Second, this method had educational value, since it encouraged the student to wisely select his frequencies to both minimize his work, and to get an accurate plot. To get 4 additional points all he had to do was re-specify the frequency. It took about 15 seconds for each set of 4 additional points. After the plots of Figures 55 or 56, he could inspect the node voltages or move on to the "selection display" of Figure 57. From this point he could rapidly analyze his network in depth. If he pressed key "Lab" and then asked for equivalent circuit analysis at f=15.95 Hz, he was shown the displays of Figure 58 and 59. If he then selected nodal analysis, he was shown the display of Figure 60. After this display, he could choose to see the component voltages, as in Figure 61.

The program had two other branch points for educational purposes. The program used nodal analysis with LU decomposition

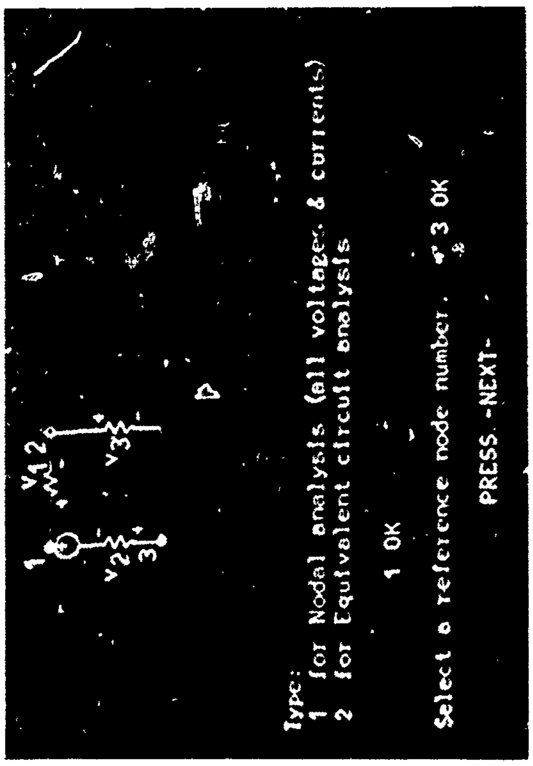


Figure 48. Analysis Specification.

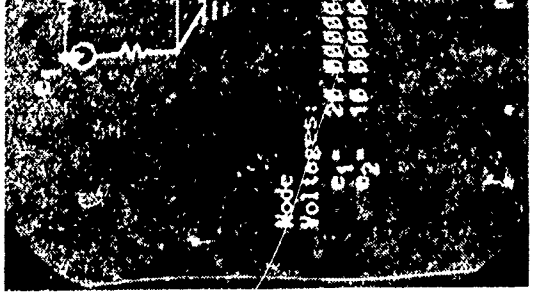


Figure 49. Node

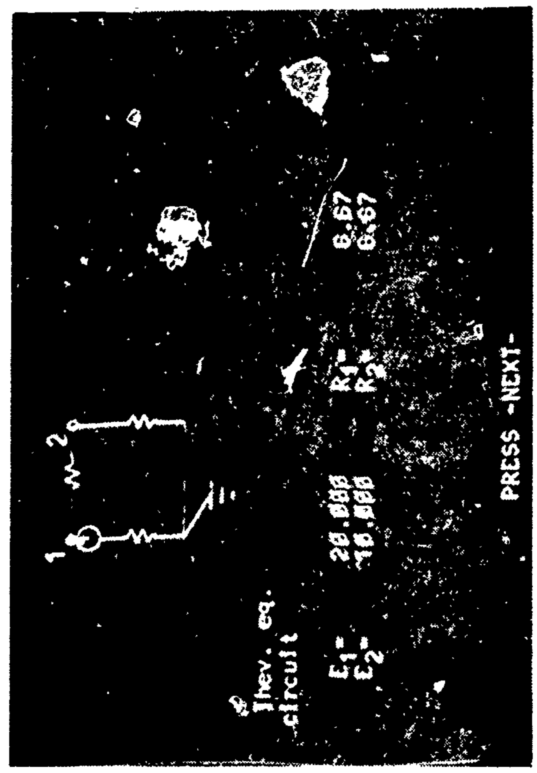


Figure 50. Thevenin Equivalent.

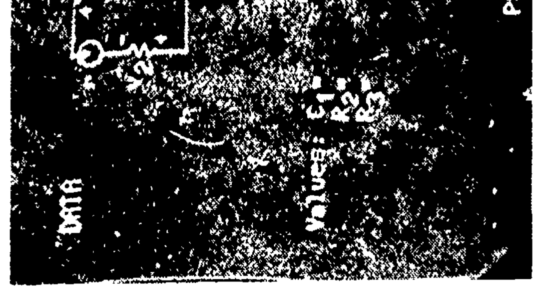


Figure 51. AC A

the available lessons. The students liked best the tutorial material on Kirchhoff's laws. They did not like using analysis and simulation routines until after they had had almost all of the theory involved. The lessons were revised as a result of the student data and comments that were collected, additional tutorial material was begun for the next semester, and additional research was done on response evaluation.

6.2 <u>The Second PLATO Section</u>

In February, 1971, the second PLATO section began with 13 registered students, of which one dropped out of the university during the semester and one changed to another section after the first week of classes. Thus 11 students completed the course; 8 had been volunteers and 3 had been assigned during registration. Thirteen of 43 class meetings, or about 30% were PLATO classes. All of the material discussed in Section 5 was used.

The programmed text by Balabanian and LePage (5) was used for the first part of the course (linear resistive networks)

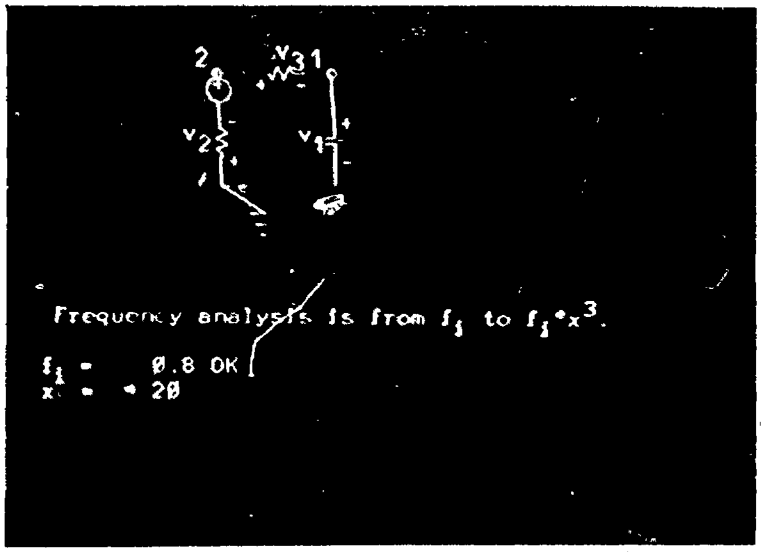


Figure 52. Frequency Specification.

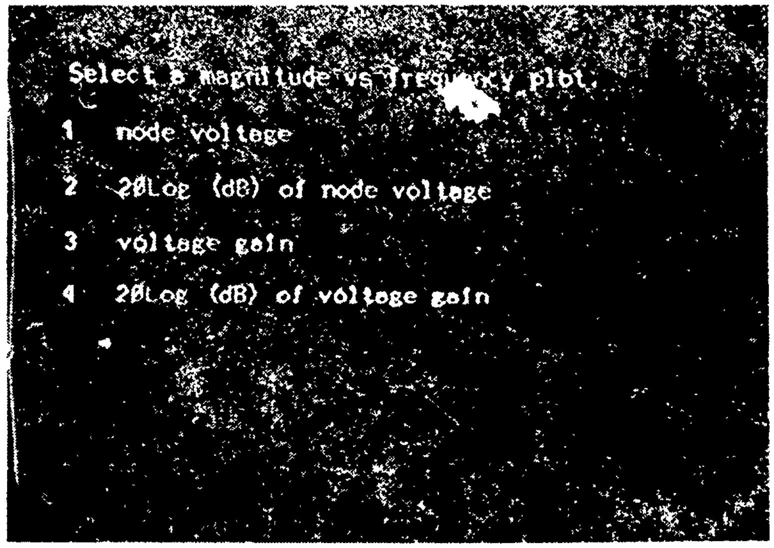


Figure 53. Frequency Plots Available.

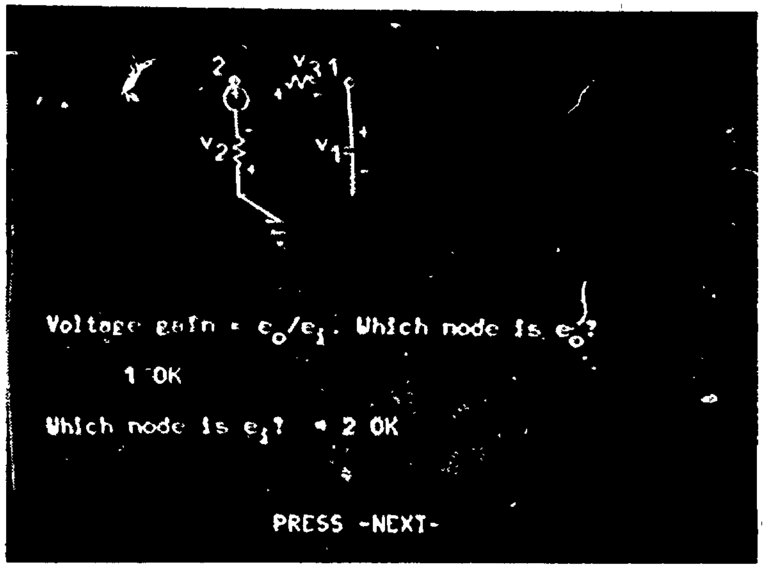


Figure 54. Gain Specification.

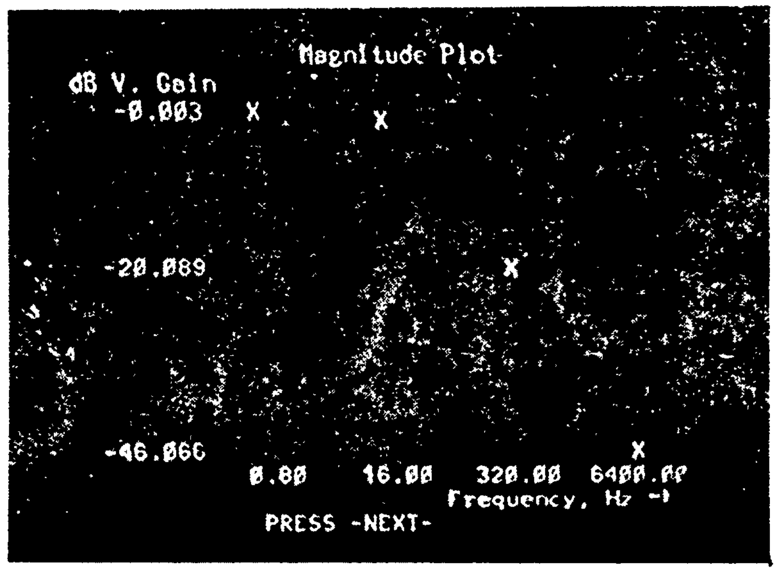


Figure 55. Log Gain Plot.

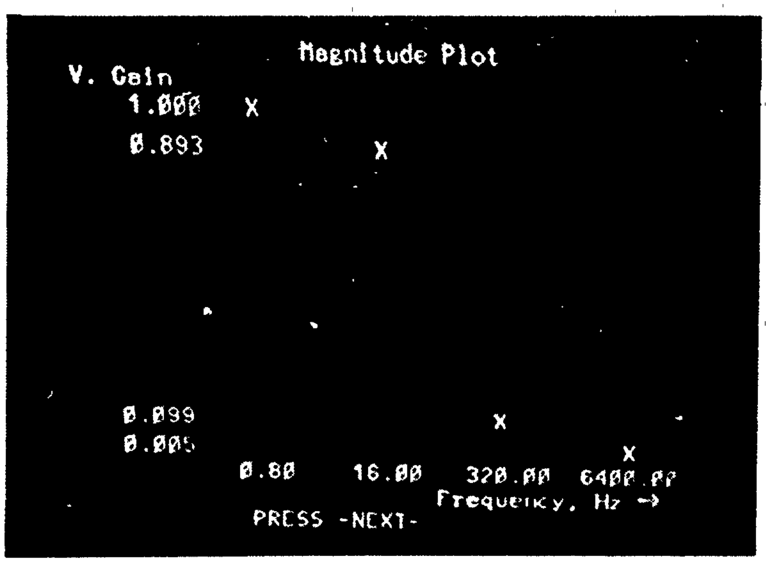


Figure 56. Gain Plot.

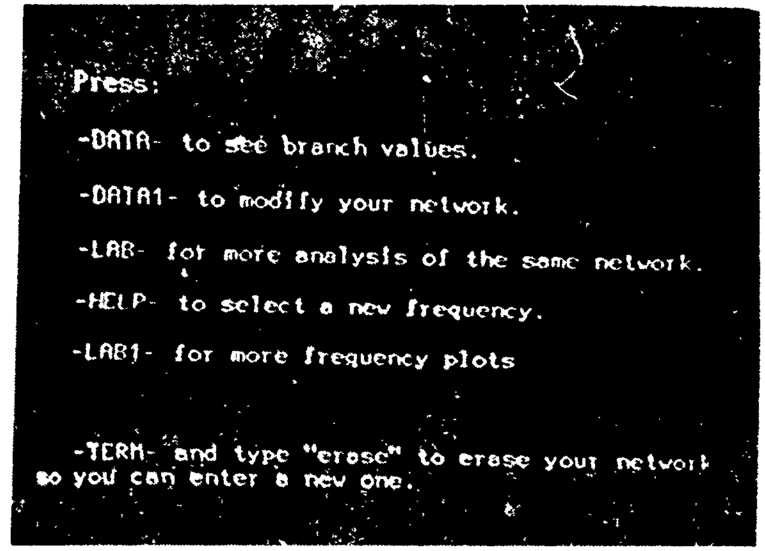


Figure 57. Analysis in Depth.

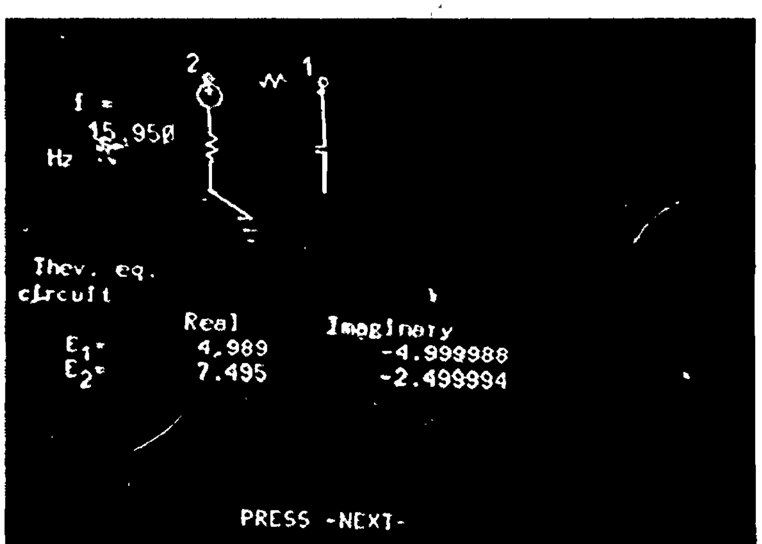


Figure 58. AC Equivalent Voltages.

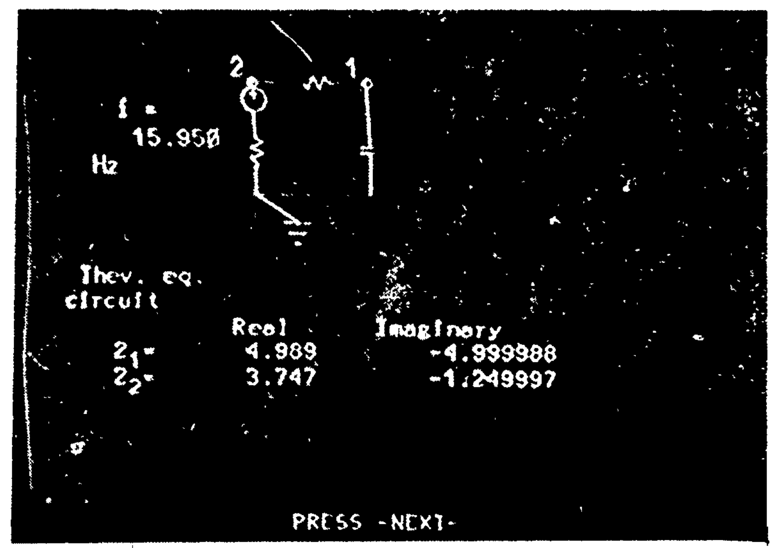


Figure 59. AC Equivalent Impedances.

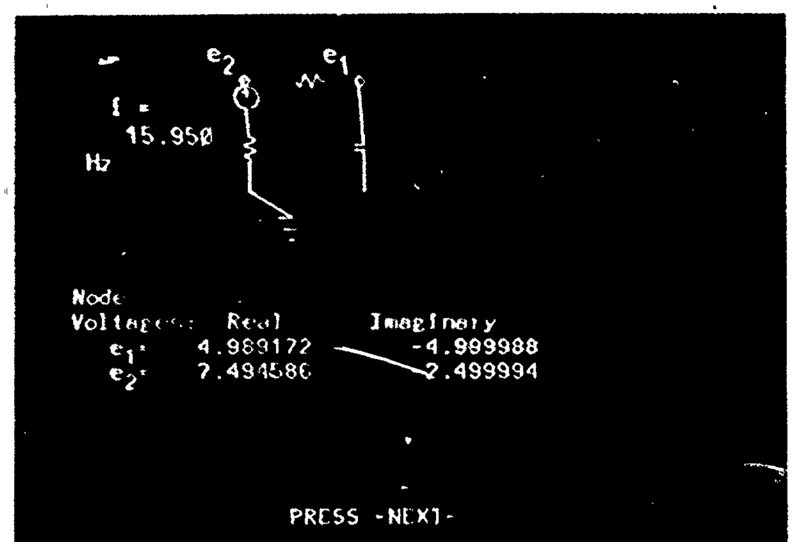


Figure 60. Complex Node Voltages

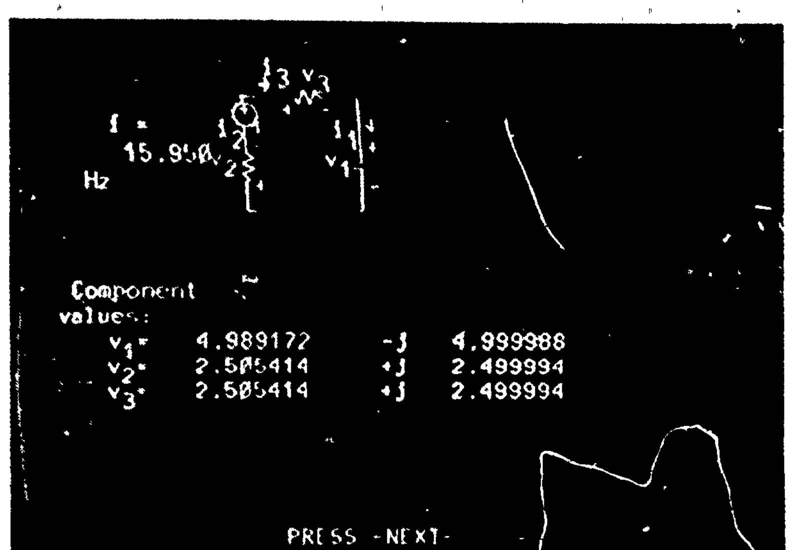


Figure 61. Complex Component Voltages

classroom. While using the analyzer in a PLATO class, he most often (but not always) had a written procedure to follow. This procedure was given to the students on the day of the PLATO class. The approach used in these procedures was to briefly mention the topic, to ask questions about it, and to suggest investigations with the analyzer. Since the first part of the course treated only linear resistive networks, the analyzer was initially restricted to DC analysis without controlled sources.

A network could be entered in less than a minute, while the solutions were available in less than 5 seconds. The ease of network entry and modification, along with the rapidity of solution caused highly favorable student reaction. The first working version of the analyzer was criticized mainly for its restrictions. The students "demanded" changes and got them. Some of the student comments on the final version were:

> It should be made available for all EE260 students.
> It lacks some ECAP capabilities, but it's easier and quicker.
> Analysis is fun on PLATO.

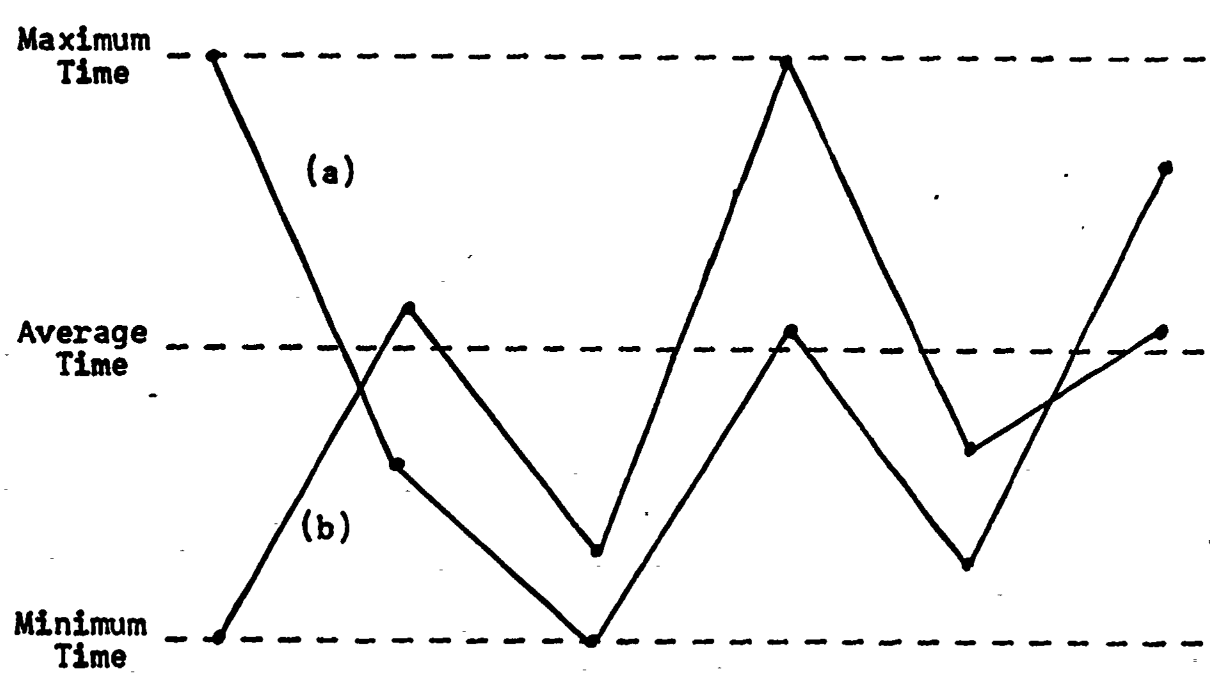
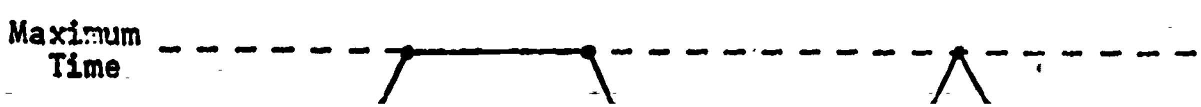


Figure 77. Lesson Completion Times of "Average Speed" Students for 6 Tutorial Lessons.



network.) The analog computer block diagram and the differential equation that was to be solved are shown in Figure 62.

Figure 63 shows the display used to enter data. As indicated in this figure, the student could specify the initial current state, i(0); the values of L and R; the vertical scaling parameter, i_{max}; the time to stop integrating, t_{end}; and the input voltage, v(t). The integration step size was computed for him. The inputs allowed were those of Figure 64. He pressed key "LAB" to obtain a plot of the complete solution for i(t). For his data of Figure 63, he obtained the plot of Figure 65. The plots were presented point by point in simulated time and were completed in about 10 seconds. Next, the student could modify the network parameters, as in Figure 66, and request a new plot as in Figure 67.

The theory was first introduced through a programmed text (25) and in the classroom via lecture, discussion and written

---

*Princeton Circuit Analysis Program (PCAP). An imprcved version of ECAP that allows plotting, among other things.

an average of 45 minutes of PLATO time to complete. This would be referred to as 3/4 of an average student contact hour. A broad estimate is that for the remaining tutorial lessons, development times ranged from about 300 hours per average student contact hour to about 100 hours per average student contact hour. These estimates include the programming time plus the time to test it on 1 - 2 students from the target population and to revise it for classroom use. The most time consuming programming element appeared to be response evaluation.

Development times for drill and practice lessons took considerably less time. A worthwhile lesson for many network theory topics could be developed in 25 hours or less. For most drill and practice lessons, the student contact time is more or less open ended. Thus student contact hour figures are meaningless.

The analysis and simulation routines took the most development time, and this time i·creased rapidly as the level of inter-

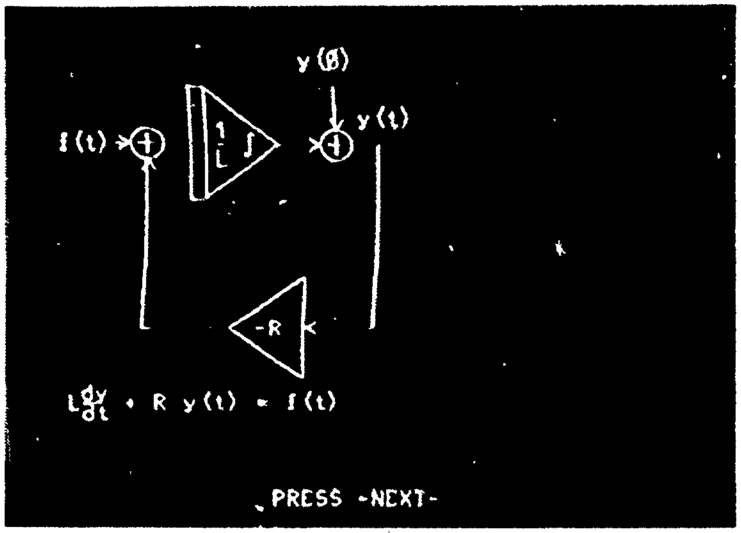


Figure 62. Block Diagram.

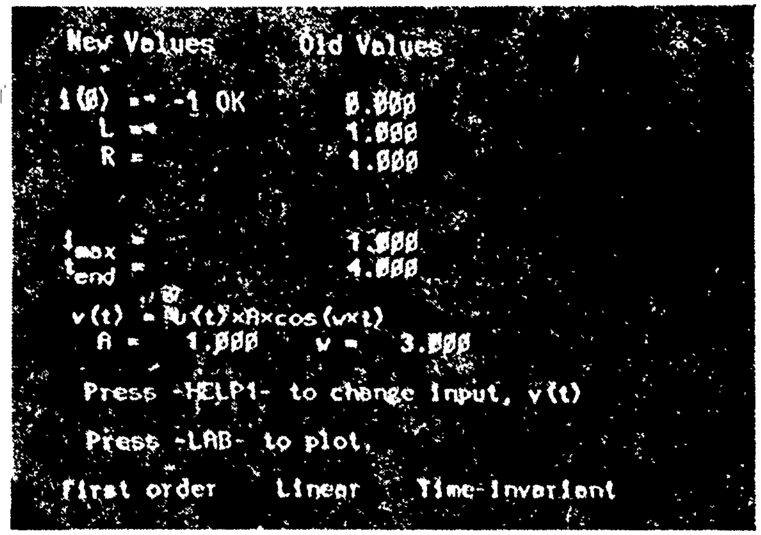


Figure 63. Data Entry.

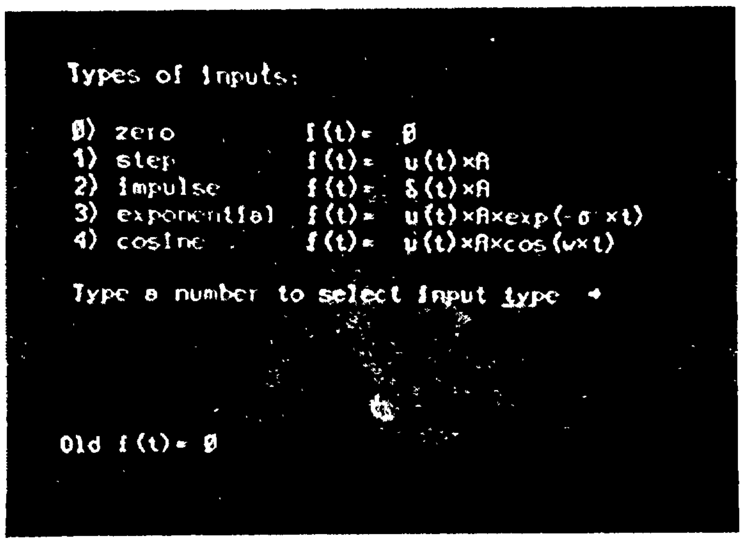


Figure 64. Allowed Inputs.

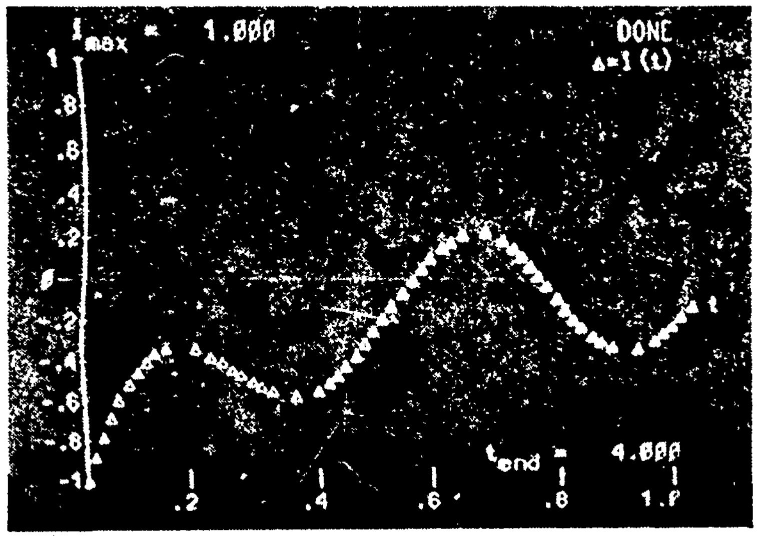


Figure 65. Numerical Solution Plot.

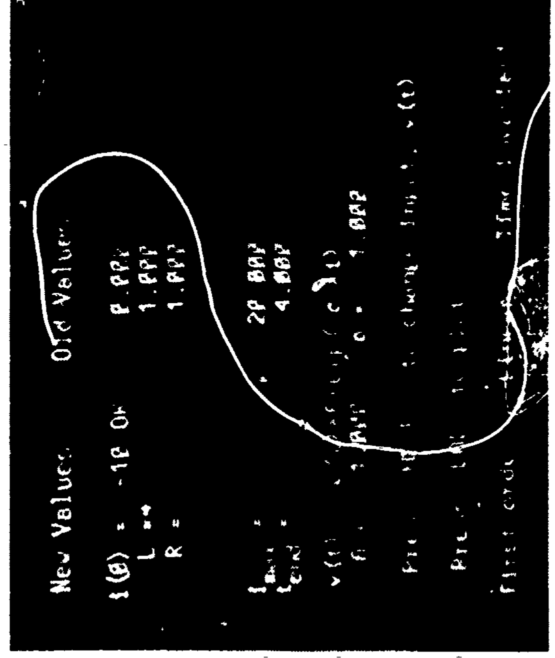


Figure 66. Data Modification.

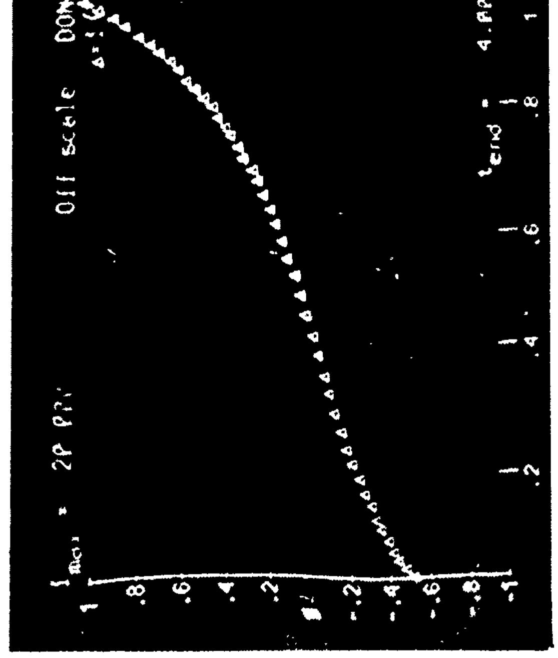


Figure 67. New Solution.

what time is taken to do a unit of group instruction, some learners will be bored and some will not be able to understand. If time is fixed student achievement tends to be the dependent variable. In the PLATO tutorial lessons, the learning was relatively fixed while the time was allowed to be the dependent variable.

A summary of the student comments produced the following PLATO utopia lesson. It was a tutorial lesson with abundant help sequences, capability for review, voluntary drill and practice, sophisticated response analysis, non-ambiguous statements and questions, ample branching on student performance, capability to ask PLATO a question, and no program errors.

Use of the analysis and simulation routines before the appropriate theory was adequately treated was found to be a mistake. The students who understood sufficient theory beforehand benefitted tremendously and enjoyed the power of the routines. But those who were still learning and assimilating the theory did

Figures 65 and 67.

5.2.7 Natural Response of a Parallel RLC Network

A program was developed to allow detailed analysis of the natural response of a 2nd order network. A parallel RLC network with fixed initial conditions was shown to the student as in Figure 68. After he entered values of C, L, and R, he was shown the important circuit parameters, as in Figure 69. An instructor-executed branch at this point could withhold the parameters until questions were asked about them. The student could then ask for a plot of the inductor current, $i_L(t)$, the capacitor voltage $v_C(t)$, and the phase-plane plot $v_C(t)$ versus $i_L(t)$. One such plot is shown in Figure 70. He could then enter a new set of values, as in Figure 71, and inspect the new plot, as in Figure 72. Again, the plots were developed point by point in simulated time.

The emphasis was on investigation of the 4 cases: lossless, underdamped, critically damped, and overdamped. Again, the theory was introduced beforehand, and values were suggested which would demonstrate each of the four cases.

were used. The first non-PLATO class turned out to be a discussion with eager, enlightened class participation. The instructor had been relieved of transmitting facts and could attempt, instead, to enrich and enliven the discussion.

During the PLATO classes, each student was fully occupied by the lesson. When there were no problems with the lessons, this researcher was free to roam about the consoles, checking student progress. He could stop by any student and make a comment, ask a question, or extend the material at hand. He had the satisfaction that he was not ignoring his other 10 students while doing so. (If he merely stopped behind a working student, that student stopped interacting with the lesson. Eight out of 11 said they often got nervous when the instructor stopped behind them. They apparently considered interaction with PLATO a personal matter.)

Two items not yet mentioned affected the outcome of this research. One is response time of the PLATO III system to

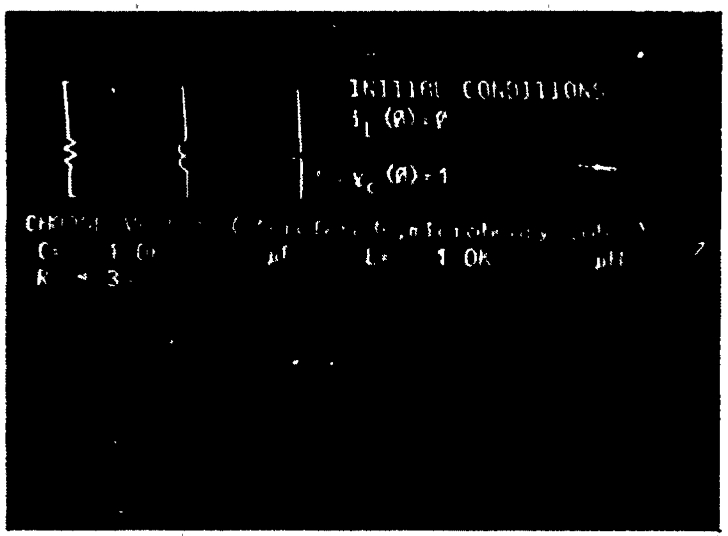


Figure 68. RLC Network.

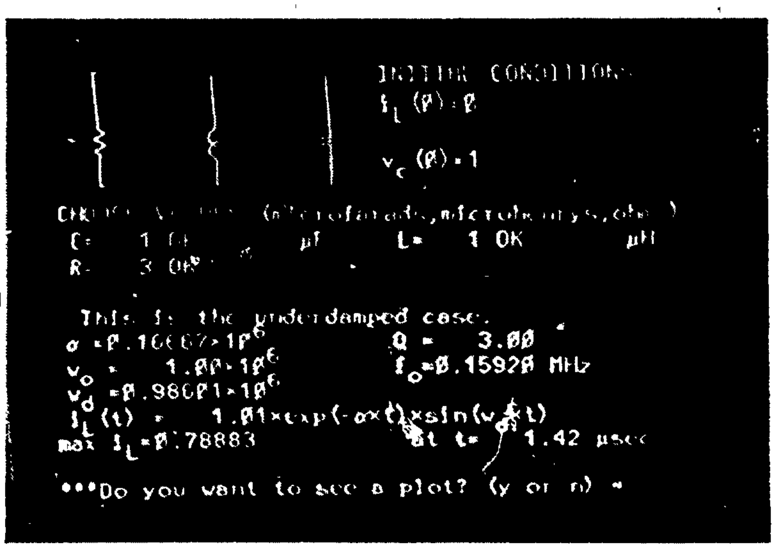


Figure 69. Underdamped Circuit Parameters.

Figure 70. Underdamped Case Plot.

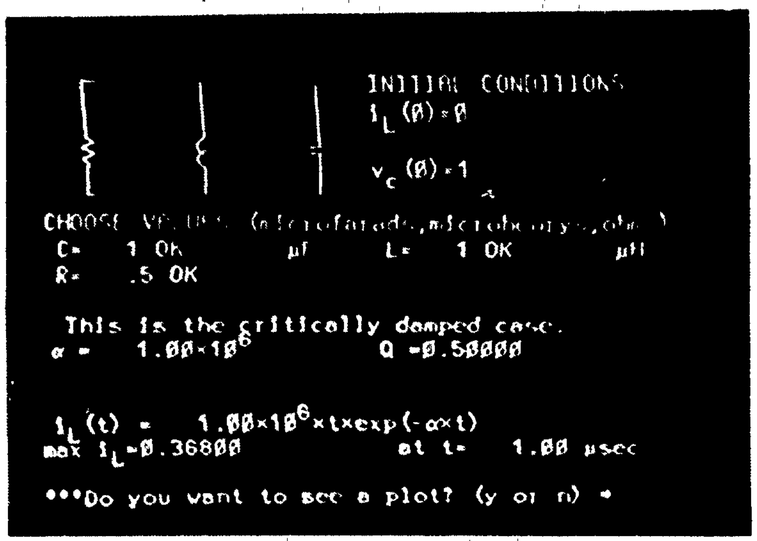


Figure 71. Critically Damped Case.

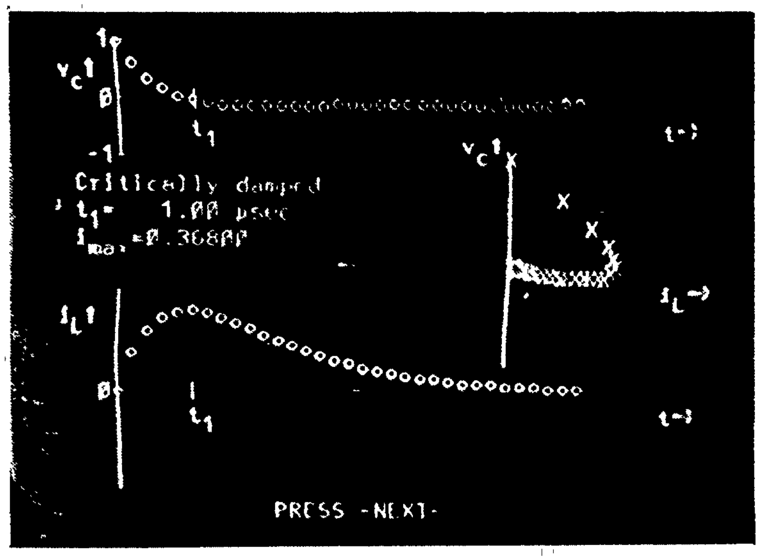


Figure 72. Critically Damped Case Plot.

5.2.8 Fourier Sums

A short program was developed to show how sums of sinusoids could be used to approximate non-sinusoidal periodic waveforms. The purpose was to motivate the student to the study of sinusoidal response.

First, he was shown how sine waves could add to approximate a square wave. This part began with the fundamental only, as in Figure 73. Each time he pressed key "NEXT", the next harmonic of proper amplitude was added, up to the 9th harmonic. Figure 74 shows the sum up to the 7th harmonic.

After this, he was shown the Fourier sum at the top of Figure 75. He could sum from 1 to 5 harmonic terms, selecting the amplitudes and phases from the A and B coefficients. The plot for the data of Figure 75 was shown in Figure 76.

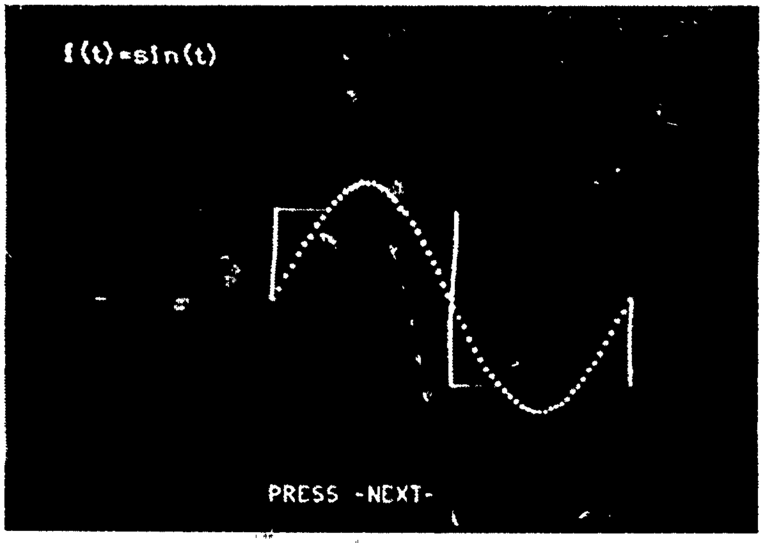


Figure 73. One Term Approximation.

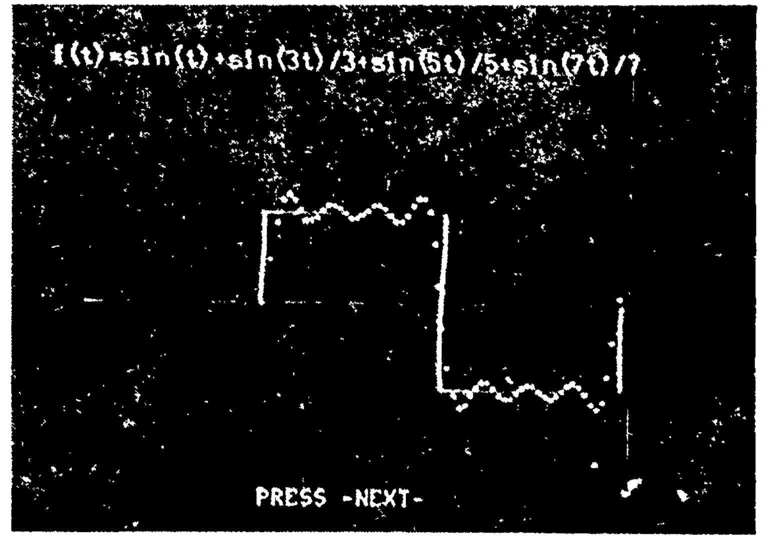


Figure 74. Four Term Approximation.

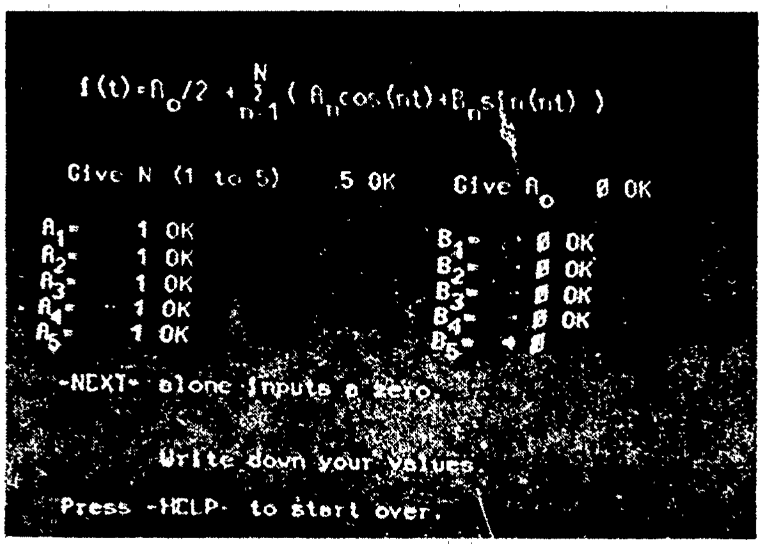


Figure 75. Coefficient Specification.

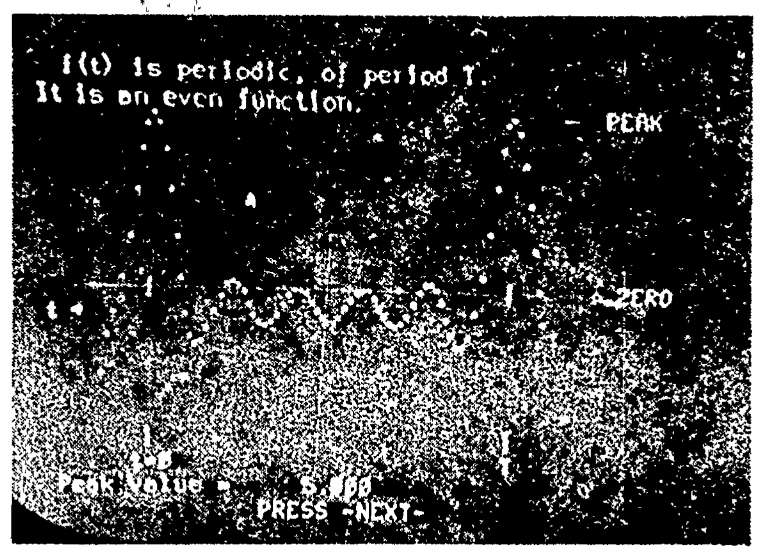


Figure 76. Periodic Function.

6. ADDITIONAL RESULTS

This research lasted for 21 months, from September, 1969 through May, 1971. The first 12 months the author became familiar with the TUTOR language and the EE260 course, developed the network drawing utility routine (Section 5.1.1), developed the lesson on Kirchhoff's laws (Section 5.2.1), and developed a DC analysis version of the on-line network analyzer (Section 5.2.5). During the last 9 months, two consecutive sections of EE260 were taught using PLATO, new programs were developed, and program revision was carried out.

6.1 The First PLATO Section

In September, 1970, the first PLATO section of EE260 was initiated with 10 volunteers registered for credit.* Out of 44 scheduled class meetings, 9 were at a PLATO console. The only tutorial material available was the introductory lesson on Kirchhoff's laws (Section 5.2.1).** The other teaching lessons available were the network analyzer (DC analysis only)(Section 5.2.5), the Dynamic Network Analyzer (Section 5.2.6) and the Natural Response Analysis lesson (Section 5.2.7). These last two were developed during the semester.

The chief results of this first PLATO section were having the author become familiar with how the student interacted with the course subject matter and having the students use and critique

---

*The section had, in addition, 2 auditors. Two of the 10 registered dropped out of the university during the semester.

**A special thanks is due here to Dr. John P. Gordon. In the spring semester, 1970, two of his EE260 sections used and critiqued this lesson.

notes. Written notes were distributed for sinusoidal steady state analysis.

Detailed results of lesson use are given in Section 5.2 for each lesson. As in the first PLATO section, students in this section liked the tutorial material, especially the lesson on Kirchhoff's laws. In contrast to the first PLATO section, however, most of the students liked the analysis and simulation lessons. Lesson revisions were made with the help of student data and comments.

6.2.1 Attendance at PLATO Classes

The average attendance of the 11 students at the 13 PLATO classes was around 93%. The average attendance at the 30 regular classes was around 85%. Only 1 student missed once during the first 7 PLATO classes.*

6.2.2 Student Attitudes

The overall attitude of the students toward use of PLATO was highly favorable. When asked at the end of the semester whether they would choose another PLATO course, all 11 students said: "Yes".

Another question asked was: "For what per-cent of a network theory course should each of the following media be used?" The average of the students' percentages are shown with the choices:

*That student excused himself for sickness and came in on a Saturday to make up the PLATO class.

47% PLATO only
20% Programmed text only
10% Lecture only
16% Discussion section only
7% Other (Films, demonstrations, etc.)

The average of the 3 non-volunteer students for "PLATO only" was 45%.

The students left the course with a favorable attitude for network theory also. Asked if they would take another network theory course, 10 selected the choice: "Yes, even if as an elective." The 11th student stated he wasn't sure since he hadn't decided on a major field yet.

As discussed above, the students' overall attitudes were favorable. They did, however, express dissatisfaction with many fragments of the lessons. Initial criticisms were directed at "inability to operate the program". Almost all problems that occurred were with parts of lessons that were being used for the first time. In a few isolated cases, small parts of lessons were used that had not been adequately checked out beforehand. For example, in two instances the response evaluation routine failed; consequently, the instructor had to "jump" the students out of the problem area. All criticisms were welcomed as they were valuable for program revision.

6.2.3 Tutorial Lesson Completion Times for the Class

Section 5.2 gives some statistics on the time it took the class of 11 students to complete each tutorial lesson. (Times were not kept for the non-tutorial lessons.) For most of these lessons, the longest completion time was about twice the shortest, giving a ratio of about 2 to 1. For one lesson, this ratio was about 8 to 1.

The average time it took the class to complete all tutorial lessons was 185 minutes. The longest time for all lessons was 253 minutes; the shortest time was 128 minutes. Thus the overall ratio of longest to shortest completion time was about 2 to 1.

6.2.4 Tutorial Lesson Completion Times for an Individual Student

An investigation was made into the individual differences of learners for two students who had completed the set of tutorial lessons near the class time average. The times they took to complete each lesson varied widely about that lesson average. The data is plotted in Figure 77 for 6 tutorial lessons. Most other students varied less about the average time. Data for the students who completed the set of tutorial lessons in the maximum and minimum time is plotted in Figure 78 as curve (a) and (b), respectively.

6.2.5 Lesson Development Times

The experience of this researcher was that the development time for lessons naturally decreased with experience but varied widely with the instructional purpose and sophistication of the lesson. The first 1200 hours of research were spent on reading the literature, becoming familiar with PLATO III and the TUTOR language, designing special characters for electrical symbols, developing the network drawing routine (DRNET), and developing the first tutorial lesson on Kirchhoff's laws. For a student who was not familiar with these laws, the lesson took

7. EVALUATION

7.1 Conclusions

The student attitudes, as indicated by their frequent attendance at PLATO classes and by their solicited and unsolicited comments, were highly favorable toward the uses of PLATO in this network theory course. Some student attitudes toward media may not be a direct measure of their educational values (11, p. 13; 16, p. 15). However, it appears that student attitudes that are related to how well they feel they learn are good measures of effectiveness. The students were asked whether they "really learned the material presented on PLATO (overall)." One student disagreed, 1 was undecided, 7 agreed, and 2 strongly agreed. (No one selected "strongly disagree".)

Educational effectiveness was measured by student achievement on written tests administered in the regular classroom. Student achievement was highest for those behavioral objectives introduced via PLATO tutorial lessons with built-in drill and practice. For example, all the students were able to "Write the Kirchhoff current and voltage law equations for arbitrary lumped networks of up to 5 nodes and 8 branches."

No control group was used and no measurements were made to compare the use of PLATO III with other media. It is hoped, however, that the results will be useful to those selecting media, at least for an introductory network theory course. (19)

Individual differences in learners appear to be manifest at as low a learning level as is investigated. The results of Section 6.2.4 and Figure 77 and 78 indicate that, no matter session. The editing features provided are outstanding. Material can be inserted, shifted, modified or deleted with the keyset. Whenever computer memory space is available, the author can compilehis materials and check them out in the student mode.

7.2 Recommendations

This research indicates that network theory can indeed be taught with computer assistance. More research is needed to further define what is educationally possible.

The TUTOR programs developed in this research should continue to be used, revised, and supplemented. The response evaluation routines should be improved, especially in providing diagnostics. Some of the tutorial lessons should have added branching on student performance and more help sequences should be provided. The analysis routines should have improved interactive capability.

The following instructional modes should be investigated: student directed inquiry, dialog, and inductive versus deductive approaches.

This research did not provide for any formal testing with PLATO, nor did it provide for interactive grading of problem sets. Both of these should be attempted.

Two other features that should be investigated are inter-station communication and also the use of pre-recorded audio messages for part of the feedback.

Finally, research should be done on the information-structure-oriented approach, which is based on the use of an information network of facts, concepts, and procedures. (8, 15, 18, 23, 24)

LIST OF REFERENCES

1. Alpert, D. and Bitzer, D., "Advances in Computer-based Education", Science, Vol. 167, (20 March 1970), pp. 1582-1590.

2. Anderson, R. C., Kulhavy, R. S., and Andre, T., Feedback Procedures In Programed Instruction, Computer-based Education Research Laboratory, University of Illinois, Urbana, Illinois, February, 1970.

3. Ausubel, D. P., and Robinson, F.G., School Learning, Holt, Rinehart and Winston, Inc., New York, N. Y., 1969.

4. Avner, R. A. and Tenczar, P., The TUTOR Manual, Computer-based Education Research Laboratory, University of Illinois, Urbana, Illinois, 1969.

5. Balabanian, N., and LePage, W. R., Electrical Science, Book 1, McGraw-Hill Book Co., New York, N. Y., 1970.

6. Bitzer, D. L., and Johnson, R. L., "PLATO: A Computer-Based System Used in the Engineering of Education", Proceedings of the IEEE, Vol., 59, No. 6, (June 1971), pp. 960-968

7. Bloom, B. S., "Learning for Mastery", Evaluation Comment, Center for the Study of Evaluation of Instructional Programs, University of California at Los Angeles, Vol. 1, No 2, (May 1968)

8. Carbonell, J. R., "AI In CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction", IEEE Transactions on Man-Machine Systems, Vol. MMS-11, No. 4, (December 1970), pp. 190-202.

9. Carroll, H., "A Model of School Learning", Teachers College Record, Vol. 64, (1963) pp. 723-733.

10. Easley, J. A., A Project to Develop and Evaluate a Computerized System for Instructional Response Analysis, Final Report, U.S. Department of Health, Education, Welfare, September 1968.

11. Educational Technology in Higher Education: The Promises and Limitations of ITV and CAI (report of the Instructional Technology Committee of the Commission on Education of the National Academy of Engineering), (1969).

12. Entwisle, D. R. and Huggins, W. H., "Simulated Environments in Higher Education", The School Review, Vol. 75, No. 4, (Winter 1967), pp. 378-391.

13. Gagne, R. M., "Some New Views of Learning and Instruction", IEEE Transactions on Education, Vol. E-14, No. 1, (February 1971), pp. 26-31.

14. Gronlund, N. E., Measurement and Evaluation in Teaching, The Macmillan Co., Collier-Macmillan Limited, London, 1965.

15. Hall, J. N., "Generation: A Computerized Tour", M.S. Thesis, University of Connecticut, 1971.

16. Jenkins, J. R. and Deno, S. L., "A Model for Instructional Objectives", Educational Technology, December 1970, pp. 11-16.

17. Koen, B. S., "Self-Paced Instruction in Engineering: A Case Study", IEEE Transactions on Education, Vol. E-14, No. 1, (February 1971)

18. Koffman, E. B. and Hall, F. N., "A Generative CAI Tutor and Problem Solver", Purdue 1971 Symposium on Applications of Computers to Electrical Engineering Education, pp. 368-375.

19. Koontz, J. L., "An Engineering Approach to Designing Instructional Systems", Engineering Education, Vol. 61, No. 6, (March 1971), pp. 528-531.

20. Mitzel, H. E., "The Impending Instruction Revolution", Engineering Education, Vol. 60, No. 7, (March 1970), pp. 749-754.

21. Neal, J. P. and Meller, D. V., "Computer-Guided Experimentation - A New System for Laboratory Instruction", Purdue 1971 Symposium on Applications of Computers to Electrical Engineering Education, pp 182-189.

22. Sherwood, B. A., Asst. Professor of Physics, University of Illinois, Urbana, Illinois, private communication.

23. Wales, C. E., "1976", IEEE Transactions on Education, Vol. E-14, No. 2, (May 1971), pp. 49-53.

24. Wexler, J. D., "Information Networks in Generative Computer-Assisted Instruction", IEEE Transactions on Man-Machine Systems, Vol. MMS-11, No. 4, (December 1970), pp. 181-190.

25. Williams, E. M. and Mukhopadhyay, A. K., Solutions of Ordinary Linear Differential Equations with Constant Coefficients, John Wiley & Sons, Inc., New York, N. Y., 1968.

VITA

Roger Louis Grossel was born in Stangelville, Wisconsin on December 23, 1934 and was graduated from St. Norbert High School, West DePere, Wisconsin in 1952. He received the B.S. and M.S. degrees in Electrical Engineering from the University of Wisconsin, Madison, in 1958 and 1964 respectively.

He was commissioned in the U.S. Air Force as a Second Lieutenant in 1958. Since then he has had several Air Force assignments, which involved instrumentation of ground radars, testing of air defense equipment, infrared techniques, and detection of ground targets.

Major Grossel is a member of the Institute of Electrical Engineers and the American Society for Engineering Educators. He is the author of "Statistical Analysis of Empirical Distribution Functions", M.S. Thesis, 1964.